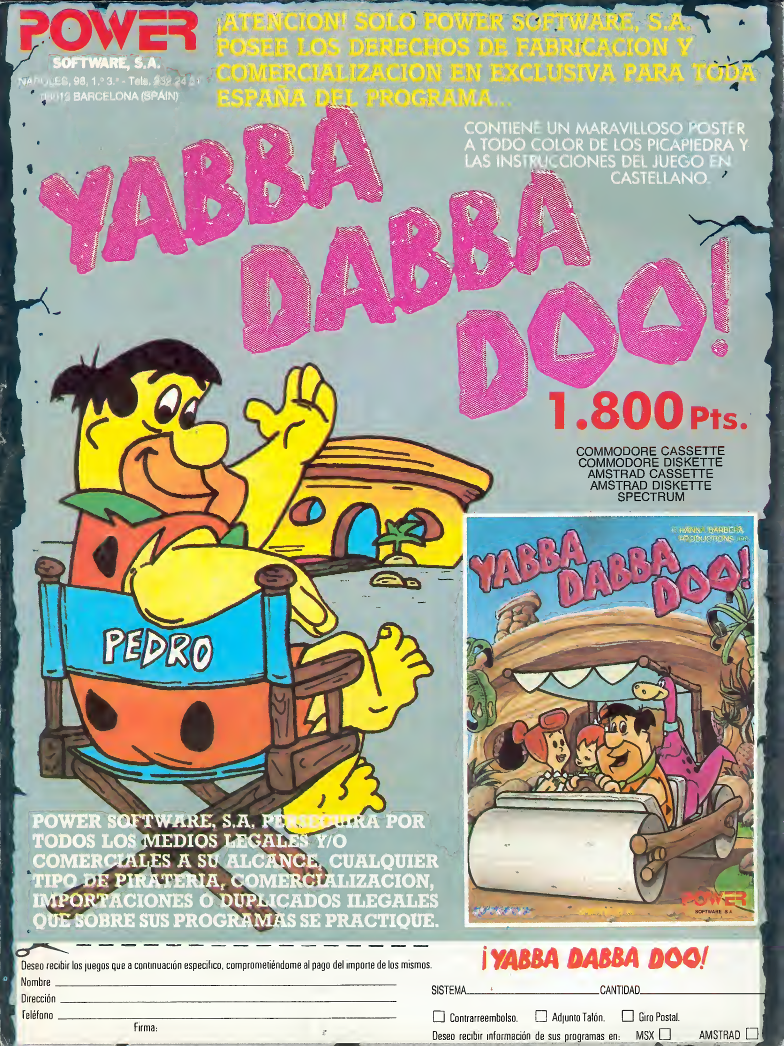


RAMBO: LA AVENTURA

SOFTMODORE PARA C-16

LOS PICAPIEDRA: ¡YABBA DABAA DOO!

PANTAMON: GENERADOR DE PANTALLAS


POWER
SOFTWARE, S.A.
BARCELONA (SPAIN)
¡ATENCION! SOLO POWER SOFTWARE, S.A. POSEE LOS DERECHOS DE FABRICACION Y COMERCIALIZACION EN EXCLUSIVA PARA TODA ESPAÑA DEL PROGRAMA...
YABBA DABBA DOO!
CONTIENE UN MARAVILLOSO POSTER A TODO COLOR DE LOS PICAPIEDRA Y LAS INSTRUCCIONES DEL JUEGO EN CASTELLANO.
1.800 Pts.
COMMODORE CASSETTE
COMMODORE DISKETTE
AMSTRAD CASSETTE
AMSTRAD DISKETTE
SPECTRUM
PEDRO
© HANNA BARBERA PRODUCTIONS, INC.
YABBA DABBA DOO!
POWER SOFTWARE S.A.
POWER SOFTWARE, S.A, PERSEGUIRA POR TODOS LOS MEDIOS LEGALES Y/O COMERCIALES A SU ALCANCE, CUALQUIER TIPO DE PIRATERIA, COMERCIALIZACION, IMPORTACIONES O DUPLICADOS ILEGALES QUE SOBRE SUS PROGRAMAS SE PRACTIQUE.
Deseo recibir los juegos que a continuación especifico, comprometiéndome al pago del importe de los mismos.
Nombre
Dirección
Teléfono
Firma:
¡YABBA DABBA DOO!
SISTEMA
CANTIDAD
Contrarreembolso.
Adjunto Talón.
Giro Postal.
Deseo recibir información de sus programas en: MSX AMSTRAD



Director:
ANTONIO M. FERRER ABELLO

Redacción:
ANTONIO CARVAJAL
JUAN M. LOPEZ MARTINEZ
PABLO GARCIA MOLINA
IGNACIO BARCO LUENGO
ALFEDO SINDIN VALERO

Colaboradores:
JOSE LUIS VAZQUEZ DE PARGA
ANTONIO MANZANERA

Secretaría de Redacción:
PILAR MANZANERA AMARO

Diseño y Maquetación:
CARLOS GONZALEZ AMEZUA
CARLOS TALLANTE PANCORBO

Ilustraciones:
ANTONIO PERERA
RAMON POLO

Fotografía:
Equipo Gálata

Directora Publicidad:
CARMINA FERRER
Tel. (91) 457 69 23

Publicidad Barcelona:
ISIDRO IGLESIAS
Avda. Corts Catalanes, 1010
Tel. (93) 307 11 13

Director de Producción:
VICENTE ROBLES

Directora de Administración:
MARIA ANTONIA BUITRAGO

Suscripciones:
MARIA GONZALEZ AMEZUA

Redacción, administración, publicidad y suscripciones:
Plza. República Ecuador, 2.
28016 MADRID. Tel. 250 58 20
Télex 49371 ELOC E

Dirección para correspondencia:
Aptdo. de Correos 61.294
28080 MADRID

Fotocomposición:
VIERNA, S. A. MADRID

Fotomecánica:
RODACOLOR, S. A. MADRID

Imprenta:
GRAFICAS REUNIDAS, S. A. MADRID

Distribución:
COEDIS, Valencia, 245.
BARCELONA

Precios para España. Ejemplar: (160 ptas. (IVA incluido). 155 ptas. (Canarias, Ceuta y Melilla). (La suscripción anual incluye 52 números.)
Distribución Cono Sur:
CADE, S.R.L.
Pasaje Sud América, 1532
Tel. 21 24 64
Buenos Aires 1.290. Argentina.

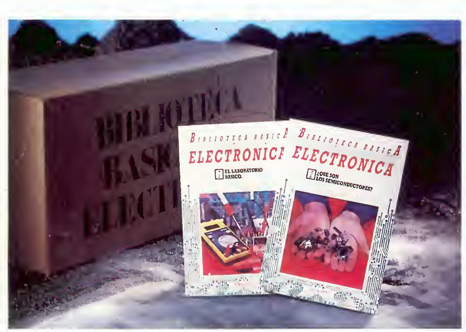
Depósito Legal: M. 40920-1985.
Impreso en España.

SUMARIO

Seguimos cumpliendo con los objetivos que nos habíamos propuesto para nuestra nueva fase, y como un buen equipo ministerial, vamos cumpliendo poco a poco, aunque sin detenernos, todas las promesas que realizamos en el número 8. Esta semana, comenzamos una serie de SOFTMODORE destinada a brindar una panorámica general de los programas destinados al benjamín de la gran familia Commodore: el C-16. ¡Tranquilizaos el resto!, que por supuesto seguís teniendo un puesto preponderante en nuestra revista. Esperamos que cada novedad que incluimos os agrade más que las anteriores.

TODA LA FUERZA DE LA ELECTRONICA

EN UNA SOLA OBRA...



BIBLIOTECA
BASICA
DE
ELECTRONICA

40 VOLUMENES TEMATICOS DE APARICION SEMANAL

Una gran obra que
pone a su alcance todo
el mundo de la Electrónica,
desde el laboratorio básico,
microprocesadores o chips,
hasta la optoelectrónica
o la robótica.



EDICIONES NUEVA LENTE

RAMBO

Los típicos sonidos de la selva (el dulce trino de los tucanes, las fieras rugiendo, las sierras eléctricas funcionando, etc.) están siendo alterados por el ruido de las explosiones y las balas. De entre el caos, un hombre solitario surge indomable: Rambo, el sustituto de superman en las selvas del Vietnam

La guerra del Vietnam representa uno de los conflictos bélicos más renombrados de este siglo. Tanto es así, que miles de libros, canciones y películas han orientado sus mensajes en torno a los combates que tuvieron lugar en el Vietnam del norte y del sur. No hace mucho tiempo, otra superproducción norteamericana salto a las pantallas de cine de todo el mundo con un gran éxito de taquilla: Rambo, cuyo argumento coincide con el de un exguerrillero, que lleva a cabo el rescate de los prisioneros de guerra norteamericanos aún encerrados dentro de las celdas de los campos de concentración vietnamitas.

Este luchador nato es el héroe indiscutible de la odisea con su potente musculatura (fruto de una esmerada alimentación macrobiótica o de la práctica asidua del aerobic) y con su buena puntería. Desde este punto de vista, podemos identificar a Rambo como una especie de Tarzán asiático que ha sustituido a Chita por un lanzacohetes, a Jane por una metralleta, y a los típicos massai por fuerzas del ejército vietnamita.

Tu serás Rambo en esta aventura y habrás de vertelas con el contingente enemigo, pues el Coronel Trautman, tu comandante en jefe, te ha ordenado claramente cuanto debías hacer: obtener pruebas fotográficas de la existencia de prisioneros norteamericanos dentro de los campos de concentración. Pero, realmente nunca te gusta obedecer, por eso has decidido fiarte de los dictados de tu espíritu aventurero y asumir por cuenta y riesgo propios el rescate de los soldados de la prisión militar.

RAMBO: EL MUÑEQUITO QUE DISPARA, CAMINA POR LA JUNGLA Y MASACRA ENEMIGOS

Así lo quisistes y nadie te ayudará ahora, debes valerte por tí mismo de principio a fin durante las tres partes integrantes de la aventura: Rescatar a tu amigo Banks, idem con los otros diez prisioneros que permanecen aún en el campo, y escapar hacia Tailandia. El enemigo, es evidente, hará todo lo posible por dificultar tu misión enviando la casi totalidad de sus efectivos en contra tuya.

Por si esto fuera poco, la jungla se caracteriza por su alto grado de peligrosidad, y sólo los escogidos sobrevivirán consiguiendo atravesarla con éxito. Multitud de obstáculos naturales del paisaje (arbustos, juncos, árboles y palmeras) surgirán inesperadamente a cada paso, con lo cual resulta conveniente practicar en un jardín próximo a nuestro domicilio antes de lanzarse a la conquista de la selva.

A aquellos lectores faltos de interés en desplazarse a un jardín cercano, les suministramos los informes botánicos de cada especie selvática (donados por el ilustre biólogo Floripondio Palmero) y la



forma de evitar sorpresas insospechadas. Por ejemplo, las briznas de hierba no interfieren en absoluto en el movimiento de Rambo, aunque todos los demás especímenes provocan su detención, salvo si decidimos pasar entre ellos (siempre que no se trate de dos palmeras situadas a muy corta distancia), o preferimos destruir un árbol o una palmera. Los proyectiles lanzados atravesarán sin dificultad cualquier barrera de vegetación, o se estrellarán conta ella arrasándola.

Un joystick indistintamente conectado al port 1 o al port 2 (ideal si no sabemos cúal es el 1 y cúal el 2), es todo cuanto necesitamos para poner coto a la situación de cautiverio de los prisioneros de guerra en el campo enemigo. Los desplazamientos del protagonista se verifican en las ocho direcciones normales del joystick, y el botón de disparo realiza la función que su propio nombre indica. Las únicas teclas hábiles son la barra espaciadora (selección del arma empleada), la «S» (faculta el escoger entre escuchar la música o los efectos sonoros), y la tecla «RUN/STOP» para tomarnos un respiro de vez en cuando, al parar el juego deteniendo el curso usual del mismo.

Sin embargo, Rambo se caracteriza por su peculiar forma de acabar con el enemigo; en este caso, miembros del ejército vietnamita, utilizando seis tipos distintos de armas: machete, flecha explosiva, flecha normal, granadas, ametralladora y lanzacohetes. Cada una de ellas con unas características indicadas para un momento determinado, de tal forma que el uso equivocado de una de ellas nos planteará nuevas dificultades y, la verdad sea dicha, no nos hace ninguna falta complicar el programa aún más (al menos, hasta efectuar a la perfección los primeros 1.000.000.000... rescates).

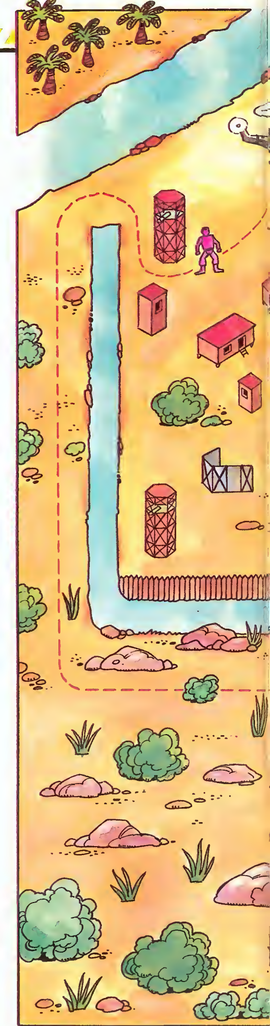


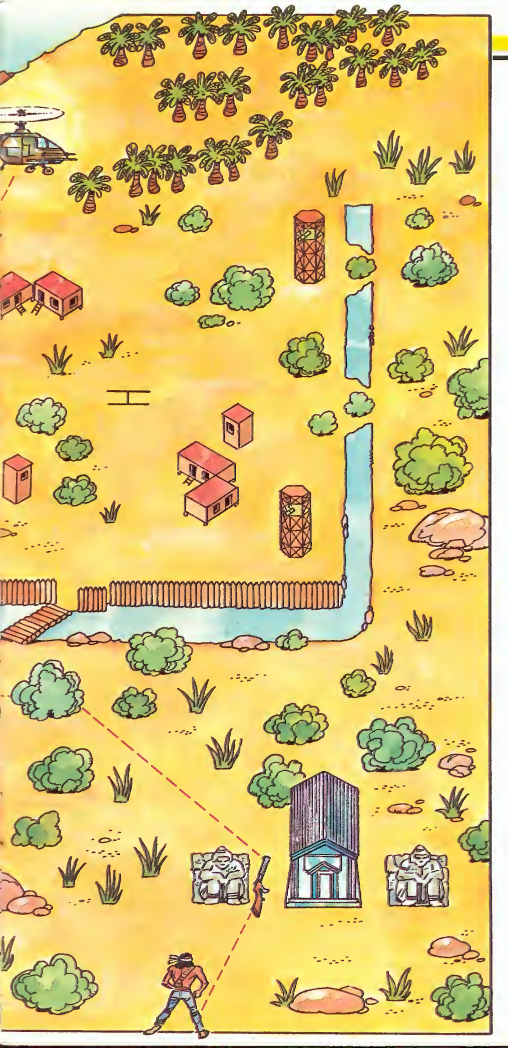




¿QUE HARIAS SI ENCUENTAS A TU MEJOR AMIGO CRUCIFICADO? LO MISMO QUE RAMBO: ACTUAR EN CONSECUENCIA

Trás leer los ochenta nombres de los autores y colaboradores del programa, y después de traducir las instrucciones que aparecen al terminar este primer proceso, tu reacción más inmediata consistiría en pulsar el disparo para definir un conjunto de caracteres dignos de ser leídos en caso de obtener una puntuación comprendida entre las ochenta mejores. Acto seguido, apretar el botón de disparo marca el comienzo de la primera parte de tu misión.





Dentro de la selva la desorientación es total, pues sólo disponemos de unos escasos metros de visibilidad y surge la duda: ¿hacia donde dirigirnos? El mapa adjunto al artículo muestra el recorrido a seguir con el fin de cumplimentar tres requisitos muy importantes: apoderarnos de la ametralladora, salvar a Banks y llegar hasta el helicóptero.

Desde el punto de partida hasta llegar a la torre superior izquierda del campo, el arma indicada es el arco, el cual nos permitirá pasar desapercibidos, sin demasiado acoso enemigo. Para liberar a Banks, seleccionaremos el machete en el cuadro de armamentos y pasaremos por encima de él. Entonces, la situación cambia y nuestra manera de actuar también lo hace, ya que trataremos de abrirnos paso entre la jungla (con la ayuda de las flechas explosivas) hasta llegar al helicóptero lo más rápidamente posible (¡lo hemos conseguido!, la primera tarea no nos dará más problemas).

LOS OTROS DIEZ PRISIONEROS DEL CAMPO ESPERAN CON ANSIEDAD TU AYUDA. ¡NO LES DEFRAUDES!

Ahora no queda más remedio que rescatar al resto de los soldados encarcelados regresando al campo de nuevo. Se advierte en el mapa el lugar donde aterrizará el helicóptero, pero el programa sólo ofrece un estrecho margen de tiempo para llegar a él. Trás tomar tierra con el aparato, Rambo irá a recoger la caja de las granadas y después, seleccionando el machete, liberará al resto de los reclusos. Durante el proceso de liberación aprovechará para escoger un nuevo armamento (lanzacohetes, ametralladora, flechas explosivas o granadas) que le permita volver sano y salvo hasta el helicóptero.









LA ULTIMA FASE, LA MAS PELIGROSA, LA MAS DIFICIL: TODO UN RETO PARA RAMBO

Despega deprisa, pon a punto el lanzacohetes y huye a toda velocidad hacia arriba, hacia Tailandia donde aterrizarás, pues el enemigo ha enviado un helicóptero ultramoderno contra ti. El proceso completo es difícil de aprender y mucho más complicado es el llevarlo a la práctica, pero una de dos: o aprendemos o nunca realizaremos un rescate al completo.

Hasta atravesar el río mantén tu aparato bajo el movimiento del joystick en la dirección diagonal arriba-izquierda (1); tres segundos más tarde, dirígete hacia la diagonal arriba-derecha y espera a que aparezca el enemigo (2), gira entonces hacia la derecha esquivando su proyectil (3), y cambia de nuevo la dirección hacia la izquierda. Repite el tercer paso y luego vuelve a comenzar el ciclo. Recuerda que el truco se basa en las posiciones de tu helicóptero: horizontal o vertical, ofreciendo un espacio mínimo para que el enemigo haga blanco sobre nosotros.

Durante todo el proceso la barra de energía de Rambo disminuirá con cada impacto del enemigo, y aumentará sobre el aparato de Rambo si este consigue alcanzar el móvil contrario (al derribarlos, el valor de la energía subirá considerablemente, y el otro aparato tardará mucho más tiempo en llegar hasta nuestra posición).

MUCHA VIOLENCIA, BASTANTE DIVERSION, ESCASA ORIGINALIDAD Y... ¡RAMBO!

Rambo no es, desde luego, lo más indicado para las personas que odian cualquier representación de la violencia. No obstante, el juego en sí ofrece mucho entretenimiento y algunas dificultades que contribuyen a darle el toque definitivo de atracción, resultando en conjunto una buena compra tanto para los jugadores novatos como para aquellos que han superado con éxito miles y miles de pantallas de toca clase.

Anecdoticamente, y con la finalidad de ayudar a los «devoradores de puntos», incluimos las puntuaciones obtenibles en el transcurso de la aventura: matar un soldado (100), coger la ametralladora o la caja de granadas (3.000 por cada una), liberará a Banks (5.000), y liberar a los prisioneros (20.000). Una puntuación muy aceptable se incluye dentro de los cuarenta primeros puestos; desde aquí, sólo necesitamos mejorar nuestras técnicas de combate hasta llegar al número uno de los cuarenta principales, ¡perdón!, de los ochenta mejores guerreros de Rambo.



NOMBRE:	RAMBO
PRECIO:	2.300 PTAS.
SOPORTE FISICO:	CINTA
TIPO:	JUEGO DE ACCION
OBSERVACIONES:	INSTRUCCIONES PESIMAS



GRAFICOS:	***
SONIDO:	***
INTERES:	****
ORIGINALIDAD:	**
DIFICULTAD:	****

el tablón

TU MICRO COMMODORE
(EL TABLON)
APARTADO de CORREOS 61.294
28080 MADRID

Os recordamos nuevamente, que ninguna carta con ánimo de lucro o que huela tan sólo ligeramente a «pirata», olor fétido de descomposición cerebral fácilmente reconocible, será publicada en nuestras páginas.

Además, aprovechamos la ocasión para haceros un ruego: escribid el texto de vuestros anuncios en un papel a parte de cualquier otro que enviéis con sugerencias, etc..., y emplead letra de imprenta lo más clara que podáis, a máquina siempre que os sea posible.

Vendo C-64 datassette C2N y joystick con seis meses de uso por 45.000 pesetas. Con la compra regalo 100 juegos y utilidades como Koala Pad, Pitfall II, One on one, Sintetizador de voz, etc. También vendería sólo el C-64 por 40.000 pesetas. Llamar de lunes a jueves, a partir de las 16 horas, al tel.: (93) 223 13 57, preguntar por Enrique Perlado Casas. C/ Rabi Ruben 6B, ático. 08004 Barcelona.

Vendo videojuegos T.R.Q.: Consola 10.000 pesetas, seis cartuchos 5.000 pesetas. Todo por 10.000 pesetas. Dirigirse a C. Urtasun. García Ximénez, 5, 1 izquierda. 31002 Pamplona.

Vendo C-64, datassette, joystick, programa recibos, programa órgano, pipeline, falcon patrol, skramble, ajedrez, donkey, kong y otros, libros, revistas, todo 50.000. Ismael Navarro Ballesteros. Cami Can Comelles, 2, 7.°, 1.ª. Esparraguera. Barcelona.

Atención usuario «serio» de Commodore 64: cambio procesador de textos Easy Script o monitor código máquina o ampliación BASIC Simon's Basic (todos con instrucciones) por una buena hoja de cálculo o contabilidad en cinta, a ser posible con instrucciones. Jerónimo Vida Manzano. C/ Santa Aurelia, 6 4° izq. GRANADA 18005. Tlf. (958) 25 25 53.

Tengo un Vic-20. Quisiera establecer contacto con otras personas que también lo tengan para intercambiar programas. José María Aragón. C/ Murillo, 7. CADIZ.

Intercambio programas para Commodore 64 en casette. Rogelio Ferrer Troyano. Avd. Reina Mercedes, 19 A. 41012 SEVILLA. Tlf. 613 37 4.

Cambio juegos y utilidades para C-64 y C-128 preferiblemente en disco. Contesto a todas las cartas. Tengo más de 700 programas. Carlos Casas Antunez. C/ Doctor Marañón 17/27 6 A. 29009 MALAGA.

Intercambio todo tipo de programas en cinta para CBM-64. Poseo últimas novedades (Nexus, Hardball, Winter, Games...). Interesados enviar lista a: Alejandro Abadía. Carretera de Daganzo, 8 6° B ALCALA DE HENARES. MADRID.

Cambio buenísimos programas por impresora, unidad de disco, joystick o monitor en color que estén en buen estado de uso. Los programas son del tipo Winter Games, Hardball, Rambo, Kawasaky, Rythm Rocker, etc... Cantidad a convenir. Interesados escribir a: José Alberto Criado. C/ Murillo, 3 1° B. ALCALA DE HENARES. MADRID.

Cambio programas para C-64 de la categoría de Blackwyche, Summer Games I y II, Zorro... También cambiaría por calculadoras programables, ordenadores de bolsillo. Cantidad a convenir. Interesados escribir a: Andrés Ramos González. C/ Eras de San Isidro, 7 1° C ALCALA DE HENARES. MADRID.

Intercambio programas en cinta para Commodore 64. Los interesados escribir a: José Maria Gómez. C/ Domingo Beltrán 1-B 1° dcha. VITORIA 01008.

Cambio 4 buenos programas por los 3 primeros números de Tu Micro Commodore. Dos programas por una revista Commodore World (1 al 13 y 15). 20 programas por libros C-64. Dirigirse a: José Ferrón González. Apdo. 209 Mérida. Badajoz. En buen estado por favor.

Cambio SPECTRUM 48k (con manuales, cables, más 300 juegos, interface Kempston, 90 revistas) por unidad de disco para CBM-64. Mi dirección: Mario Castells. Valencia, 17. Tel.: (93) 432 19 43. 0815 Barcelona.

Deseo intercambiar programas, tengo unos 180, entre ellos Ghostbusters, Pit stopll, Logo, etc., en cinta, enviar lista a: Andrés Ortega. Fidel Sarel, 13, 1.° izqu., 1.ª. 50013 Zaragoza.

Me gustaría comprar ordenador C-64 por menos de 25.000 pesetas. Mandadme lista. Joaquín Roca i Mallarach. Clot Rosat, 3. Olot. Gerona. Tel.: 26 03 06.

Atención oferta cambio por Commodore 64 y Datassette en buen estado, emisora de radio aficionado Maxcon CB 40 canales, móvil o fija, antena magnética para automóvil, idem. fija, 40 metros cable blindado antena, fuente alimentación, medidor de estacionarias, manual fácil de radio aficionado. Interesados llamar al tel.: (93) 421 76 01, o escribir a Antonio Lázaro. Iberia, 9. 0814 Barcelona.

No soy «pirata», pero por razones económicas vendo programas que ya no utilizo, así como un juego de luces rítmicas (cuatro canales), una emisora de FM (3W) y algunas cosas más (programas de Commodore-64). Tel.: 472 03 99 Madrid. También deseo contactar con programadores amigos del C-64.

Vendo VIC-20 con fuente de alimentación, unidad de ciclo y pantalla, con libros de programación, todo muy nuevo, sin usar (2 semanas); a buen precio. Harald Schillarg. C/ Cuatro 12, Castelldefels. (Barcelona). Tfno. 93-665 24 33.

¿Quieres participar en un club de estudiantes de B U.P que crea sus propios programas para ayudarse en los estudios? Escribidnos a C.E.B. CO-64. C/ Virgen de los Milagros 96, Puerto de Santamaría. (Cádiz).

Tengo un VIC-20. Quisiera establecer contactos con otras personas que también lo tengan, para intercambiar programas. Pedro Simona. C/ Portal de Cerdana. Seu de Urgen (Lérida).

Intercambio programas en cinta para Commodore 64. Mandad lista, prometo contestar. Cristóbal Dávila Frances. C/ Odena, 36, 3.°. Igualada (Barcelona).

Atención commodorianos, desearía intercambiar programas para C64, juegos y utilidades. Contesto a todas las cartas. Albert Casellas i Tusal. C/ Carme 9. Igualada (Barcelona).

Vendo VIC-20, fuente de alimentación, dos cartuchos, radar ratace, Sargon II Chess, dos cintas de iniciación, tres libros de programación, varias cintas de juegos, todo por 20.000 Ptas. Preguntar por Manolo. C/ Tenerife 27, Entresuelo 1. Tfno. 219 72 07. Barcelona 08024.

Si quereis intercambiar programas, experiencias, trucos, proyectos e iniciativas para C-64, escribir al Club Intersoft. Apartado de Correos 27016-08020 Barcelona, o a Manuel Arroyo, C/ Maresma 266, Barcelona. Entra en el primer Club de intercambio de iniciativas y programas para el C-64.

Vendo VIC-20, ampliación 16 K, tarjeta de carga rápida, libros: guía del usuario, guía de referencia del programador, curso de BASIC en dos tomos. Llamar por la noche al 254 80 62 de Madrid.

Intercambio programas para C64 en cinta; juegos y utilidades (K. Pad Dambusters, Entombed, Superzaxxon,...).
Contesto a todas las cartas. Escribir a Juan Fernando Uso Cervera. P/ Juan Bautista Tejedo, n.° 3. Burriana (Castellón).

Intercambio programas para el CBM-64 por otros de gestión o juegos. Escribid a Jesús Blanco Garcillán. C/ Calasparra, 23. 28033 MADRID.

Desearía intercambiar todo tipo de programas para CBM 64. Manda tú lista, prometo contestar. Santi Espinet Mendiola. C/ Estanislao Figueres, n.° 8, 2.° 1.°. Terrassa BARCELONA.

LOS 7 MAGNIFIC

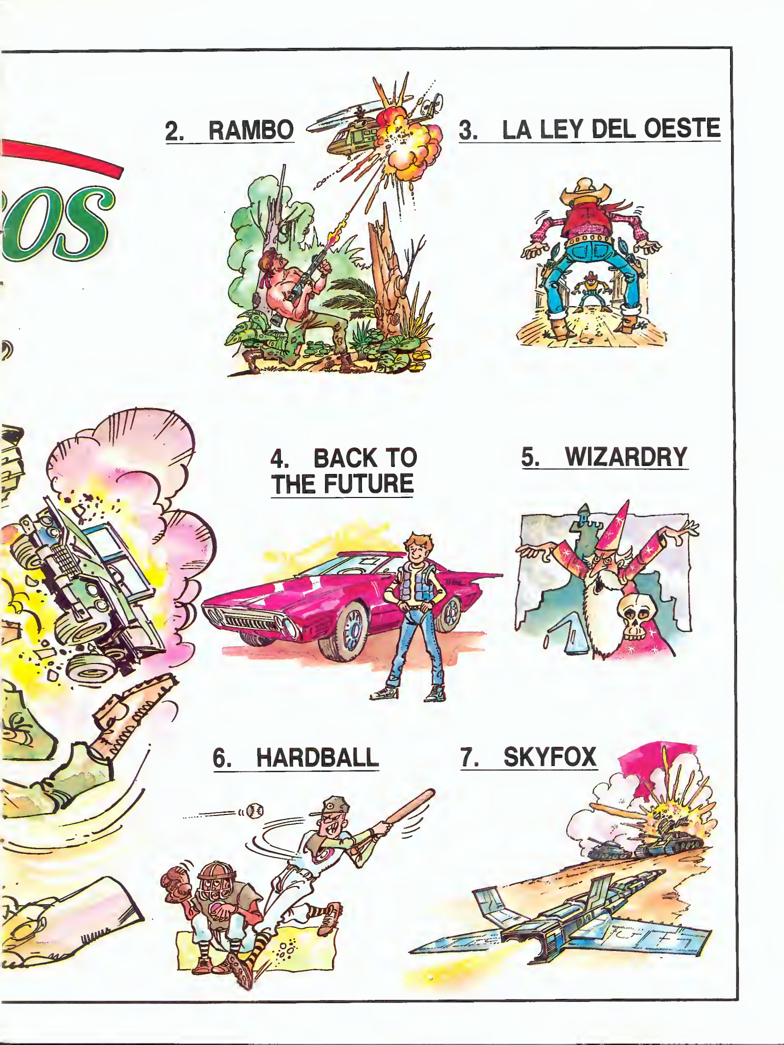
Colaboremos todos en la confección de estos 7 MAGNIFICOS de TU MICRO COMMOROE. Envíanos el nombre de tu programa favorito, dejando muy claro tu nombre y dirección. Todos los meses, sortearemos una suscripción por un año a nuestra revista entre las respuestas recibidas. Anímate y escríbenos a: TU MICRO COMMODORE. (7 MAGNIFICOS). APARTADO DE CORREOS 61.294. 28080 MADRID.



N.º 1 ESTA SEMANA

COMMANDO

El enemigo es muy superior en número y fuerzas a nosotros, pero no cuentan con nuestro excepcional arrojo y valentía, además de nuestra mayor movilidad al no ser un grupo de fuerzas, sino únicamente un solo hombre. Abandonados a nuestra suerte en medio de las líneas enemigas, para acabar con éxito nuestra misión, consistente en atravesar sus defensas lo más lejos posible, con el objeto de ir abriendo paso a nuestros aliadods. Para ello, habrá que destruir a nuestros oponentes, los cuales nos atacarán desde los lugares más inverosímiles.



2. RAMBO
3. LA LEY DEL OESTE
4. BACK TO THE FUTURE
5. WIZARDRY
6. HARDBALL
7. SKYFOX

MODOS DE DIRECCIONAMIENTO DEL 6502 (III)

Seguimos hoy analizando los modos de direccionamiento del microprocesador-cerebro de nuestro Commodore.

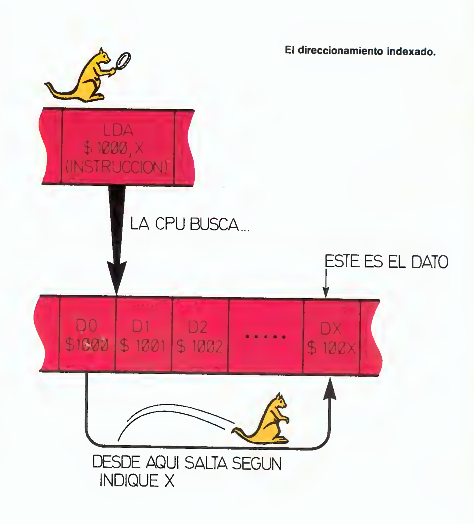
El direccionamiento indexado se utiliza para el manejo de listas de datos, es decir, de valores que se hallan en la memoria colocados de forma ordenada, unos tras otros, a partir de una dirección concreta. Hay nada menos que seis variantes de este modo, pero el funcionamiento básico es el mismo para todas ellas. Si comprendemos éste, habremos triunfado en esta batalla.



LOS REGISTROS X E Y DEL MICROPROCESADOR COBRAN AHORA SU IMPORTANCIA

Recordarás que, cuando hablamos de los registros internos de la CPU, vimos la existencia de dos registros índice, el X y el Y, que se utilizaban en las indexaciones. Pues bien, estos registros son necesarios para el direccionamiento indexado. ¿Pero qué es una indexación? Veámoslo con un ejemplo.

Supongamos que en la memo-



El direccionamiento indexado.

ria, y a partir de la dirección $1000, tenemos una serie de datos de un byte de longitud cada uno. Para poder tratar estos datos, hemos de cargarlos uno a uno en el acumulador.

Una forma muy cómoda de hacerlo sería indicarle a la CPU la dirección donde comienza la lista de datos y darle, además, de algún modo, el número del elemento que queremos cargar.

Si la CPU pudiese comprender nuestro lenguaje, le diríamos algo así: «Coge, de los bytes que empiezan en $1000, el quinto.»

Bien, pues esto es una indexación: Indexar es señalar qué elemento de la lista queremos procesar. El número de orden del elemento se entrega en uno de los registros índice. Si realmente una imagen vale más que mil palabras, tal vez la figura te pueda aclarar esta idea.

El nombre de indexación viene del hecho de emplear un índice, es decir, un registro que señala la posición del dato desde cierto punto de referencia, que es la dirección inicial.

LOS DIRECCIONAMIENTOS QUE EMPLEAN INDEXACION NECESITAN DE LOS REGISTROS INDICE

Los modos de direccionamiento que emplean indexación no sirven para controlar el flujo de programa, sino solamente para la búsqueda de datos en la memoria. Antes de emplear una instrucción con un direccionamiento de este tipo, hemos de asegurarnos que el registro índice (X o Y) que vamos a usar se halla cargado con el valor correcto.

El direccionamiento absoluto indexado es el más sencillo. Simplemente, indicamos de forma absoluta (recordar direccionamiento absoluto) la dirección de inicio, y en el registro X (o el Y) el valor que deseamos. En ensamblador, se escribe de la forma:

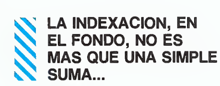
```
LDA $1000,X
```

para el direccionamiento absoluto indexado por X,y

```
LDA $1000,Y
```

para la indexación por Y.

Una instrucción de este tipo ocupa dos bytes, uno para la propia instrucción y los dos restantes para la dirección absoluta (que, como siempre, también se hallará invertida, según vimos). El uso del registro X o Y ya va incluido en la instrucción.



LA INDEXACION, EN EL FONDO, NO ES MAS QUE UNA SIMPLE SUMA...

En este modo de direccionamiento, los dos registros X e Y se comportan de forma idéntica. Esto no ocurrirá en otras variantes, como ya veremos, en las que cada registro se especializa en una tarea distinta.

En realidad, cuando la CPU encuentra una instrucción con direccionamiento absoluto indexado, lo que hace es sumar el contenido del registro en cuestión a la dirección que le hemos dado; de este modo, obtiene la dirección absoluta del dato y puede acceder a él por el bus de direcciones.

Existe también el direccionamiento absoluto indexado en página cero, cuando la dirección de inicio de la tabla es uno de los 256 primeros bytes de memoria. Es virtualmente idéntico al direccionamiento normal, con la ventaja de ahorrar un byte de memoria y un ciclo de reloj, que nunca viene mal. En ensamblador, la representación será:

```
LDA Z $FO,X
```

o bien...

```
LDA Z $FO,Y
```

si empleamos el registro Y.

Como los registros X e Y son de ocho bits, las listas que podemos tratar de esta forma deben tener un máximo de 256 bytes de longitud. Se pueden manipular listas más cortas, pero nunca listas de más de 256 bytes.

Esto puede parecer una limitación, pero la experiencia demuestra que es suficiente en la mayoría de las ocasiones. Si, de todos modos, son necesarias listas de más de 256 bytes, siempre podemos preparar un programa para tratarlas.

Veremos algunas soluciones a este problema en los ejemplos prácticos.

HOT LINE

PREGUNTA

Enrique Martín Andújar, desde Valencia, nos escribe solicitando respuesta a dos preguntas sobre su ordenador. En primer lugar aprovechamos esta carta para recordaros que en cualquier consulta que nos hagáis, indiquéis siempre toda la información que podáis, para que podamos ayudaros mejor (¡Cbyte no es adivino!), y entre otras cosas, decidnos cúal es el MODELO de vuestro Commodore.

RESPUESTA

En tu Caso, Enrique, no nos indicas esto último, así pues, te contestaremos suponiendo que se trata de un C64.

Respondiendo a tu primer problema, acerca de las interferencias que afectan a tu aparato en la emisión de sonido, por cierto, algo difícil de solucionar, vamos a efectuar ciertos pasos generales para ver si se puede corregir; si ya has dado alguno de estos pasos, por favor discúlpanos.

En primer lugar, prueba a sintonizar correctamente el canal de la TV (no hagas nada si se trata de un monitor).

Pon el volumen a un nivel medio-bajo, puesto que con niveles altos, todos los C64 se comportan de igual manera a la que nos indicas en tu carta. Lo hemos comprobado.

Si has abierto el C64, cerciórate que la cubierta de papel aluminizado está correctamente colocada. Esto puede ser origen de interferencias.

Prueba a reordenar las posiciones de los aparatos circundantes al ordenador, la TV, una lámpara de mesa, una radio, calefacción eléctrica, etc... Cualquiera que estos dispositivos puede causar problemas de ese tipo.

Comprueba si lo hace siempre, o por el contrario, sólo a ciertas horas (horas de emisión de TV regional, por la noche, hora en que se afeita el vecino, etc.); si así fuese, el problema lo podría causar una de estas cosas y no hay prácticamente solución.

Si con los juegos no lo hace, podría suceder que quedase enmascarado por los sonidos de éste. De todas formas, puedes comprobar si se debe a la TV o al ordenador cambiando de TV momentaneamente, o «apagando» el chip de sonido cuando se presente el ruido con: **POKE54296,0.**

En segundo lugar, nos preguntas si es posible conectar al ordenador alguna grabadora que no sea la de Commodore. *La respuesta es sí. Pero hemos de puntualizar a qué nos referimos con otra grabadora, ya que, directamente, sólo es posible conectar grabadoras de otras marcas que emulen a la destinada para tal fin por Commodore. En cuanto a la conexión de otra grabadora diferente (por ejemplo, el típico radio-casete), también es posible, pero hay que construirse un «interface» adecuado. Si prefieres esta última solución, te remitimos al número 10 de nuestra revista, donde hallarás todo el material necesario y cómo desarrollar este «interface».*

Que tengas suerte, y se solucione todo con algo de lo aquí expuesto.

PREGUNTA

De una forma muy escueta, Jorge de Vinaroz (CASTELLON) nos pregunta: ¿Por qué unos programas se cargan viéndose durante la misma pantalla de color azul claro y otros viéndose en la pantalla muchas líneas de colores?.

RESPUESTA

Pues bien, el proceso de carga normal de Commodore es sólo uno: el de la pantalla de un sólo color (azul claro o no, depende del color del borde en ese momento. No obstante, este proceso es sumamente lento para programas de cierta longitud (juegos y aplicaciones comerciales), por lo que, los fabricantes de software optan por crearse sus propias rutinas de carga; es el proceso conocido por «turbo».



Estos sistemas turbo, pueden ser de diferente índole y no tienen nada que ver con el ***LOAD*** *original, por lo general, se suprimen ciertas comprobaciones anti-errores, haciendo que la velocidad se pague con la falta de seguridad en la carga.*

Concretando sobre la pregunta, estas rutinas suelen permitir la visualización de la pantalla durante la carga, aprovechándose esta circunstancia para situar en ella gráficos y dibujos de alta resolución, contadores de bloques y tiempo, «scrolls» de todo tipo, y en algunos casos, por el paso de los bytes cargados por la zona de memoria de borde a fondo; como esto se produce a mucha velocidad, el cambio de color no se produce del todo cuando ya ha cambiado a otro color, resultado: aparecen unas rayas de colores cambiantes durante la carga. El sistema «NOVALDAD», hace pasar los primeros bytes por el chip de sonido, produciéndose además unos «chirridos» característicos.

CURSO DE BASIC

SENTENCIAS COMPLEMENTARIAS DE PROGRAMACION. LA SENTENCIA REM

No hemos hablado todavía de la palabra clave BASIC **REM,** cuyo significado en inglés es el de comentario (REMark), no siendo por lo tanto, una sentencia que tenga por objeto ser ejecutada por el ordenador, sino servir de ayuda para la inclusión de comentarios clarificadores de lo que el programa hace en determinados lugares, o simplemente para indicar el destino del programa al comienzo del mismo.

Todo lo que se encuentra a la derecha de la sentencia **REM** no es admitido como instrucción por el intérprete del BASIC. Debido a esta particularidad es una técnica de uso común entre los programadores anteponerla a una sentencia que deseamos que el ordenador no considere, sin suprimirla definitivamente del programa.

Cuando estamos poniendo a punto determinado programa, y pensamos que sería necesario eliminar una instrucción, pero no estamos completamente seguros de que con ello vayamos a solucionar nuestros problemas, antes de suprimir ésta, podemos optar por anteponerle un **REM.**

Hecho esto, es posible ejecutar el programa y comprobar sus resultados en la seguridad de que, si las cosas marchan correctamente, nos bastará con borrar de forma definitiva la instrucción tecleando su número seguido de **RETURN** como ya hemos visto con anterioridad.

Si, por el contrario, la instrucción debe permanecer aún en el programa por no ser la causante del mal funcionamiento del mismo, no tendremos que volver a teclearla como en el caso de que la hubiéramos suprimido desde un principio, sino que nos bastará con desproveerla de su **REM** para que el programa vuelva a quedar en las mismas condiciones de comienzo. Podemos decir, por lo tanto, que **REM** tiene el efecto de «anular» las instrucciones a las que se antepone. Veamos un ejemplo de su empleo:

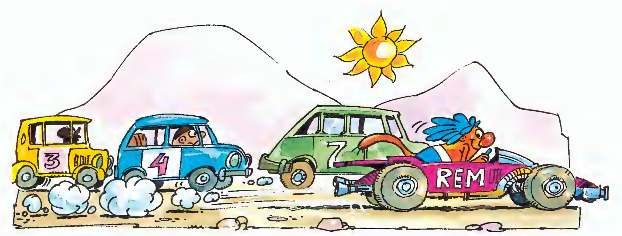
```
10 REM – CALCULO DE NOTA
MEDIA
20 PRINT ",{ CLR } "
30 INPUT "NUMERO TOTAL";N:
PRINT
40 INPUT C
45 REM PRINT C
50 T=T+C:I=I+1:IF I<N THEN
   40
60 PRINT:PRINT "NOTA MEDIA
...:"T/N
```

La línea 10 incluye un **REM** indicativo de la función que realiza el programa. En la 20 se procede a borrar la pantalla para, a continuación, aceptar por **INPUT** el número total de elementos a incluir en la media.

Concluida esta fase inicial, pasamos a la zona central del programa, comprendida en las líneas 40 a 50, las cuales aceptan los sucesivos valores de cada nota acumulándolos y controlando que su número no exceda del especificado al comienzo del programa,

La palabra clave BASIC **REM** sirve para la inclusión de comentarios en un programa.





REM *tiene el efecto de anular las instrucciones a las que se antepone.*

ejecutándose esta secuencia de instrucciones una y otra vez hasta que se produzca esta circunstancia.

Por último, en la línea 60, se imprime el resultado final dividiendo el total acumulado **T** por el número de notas **N**.

Una buena norma a la hora de confeccionar programas es «documentarlos». Para ello, podemos incluir sentencias **REM** como cabecera de cada bloque de líneas con determinado cometido, denominadas también «rutinas». De esta forma, se hace más fácil el seguimiento del programa sobre todo para aquellas personas que no han intervenido en el proceso de creación del mismo.

Como complemento de esta documentación, y llegando a un nivel mucho más descriptivo, puede incluirse al final o al principio del programa un detalle de los nombres de variables BASIC empleados y su cometido dentro del mismo. Aunque no estrictamente necesario, es cierto que facilita en gran medida el seguimiento del programa, ya que la escasez de recursos a la hora de dar nombre a las variables no permite que su nombre identifique muy claramente su función dentro del programa. Veamos un ejemplo:

```
10 REM – CALCULO DE NOTA MEDIA
11 I=0:REM CONTADOR DE ELEMENTOS
12 N=0:REM NUMERO TOTAL DE ELEMENTOS
13 T=0:REM SUMA DE ELEMENTOS
20 PRINT " {CLR} "
30 INPUT "NUMERO TOTAL";N:PRINT
40 INPUT C
45 REM PRINT C
50 T=T+C:I=I+1:IF I<N THEN 40
60 PRINT:PRINT "NOTA MEDIA...:"T/N
```

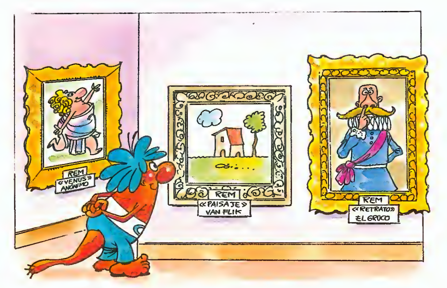
LA SENTENCIA CLEAR

Hemos adquirido ya alguna práctica en la confección de programas, familiarizándonos con la sentencia **RUN** que los pone en funcionamiento. La palabra clave **RUN** ejecuta el programa BASIC contenido en memoria a partir de la primera línea de su texto, salvo que especifiquemos otra línea de comienzo de ejecución en la forma general:

RUN línea-comienzo

La ejecución de una sentencia **RUN** implica en realidad la realización de dos funciones más elementales. La primera de ellas es la inicialización o «puesta a cero» del área de variables BASIC, al tiempo que se inicializa el puntero de lectura de **DATA** (equivalente a **RESTORE**); y la segunda el comienzo de la ejecución del programa a partir de la línea indicada, o desde su principio por defecto. Por ello, podemos decir





Una buena norma a la hora de confeccionar programas es documentarlos mediante sentencias **REM.**

que **RUN** equivale a una sentencia **CLEAR** más una **GOTO.**

La sentencia **CLEAR** está también relacionada con la palabra clave **NEW**, siendo en cierto sentido, una restricción de esta segunda. Esto es debido a que **CLEAR** no afecta en absoluto a la zona de texto BASIC, mientras que **NEW** significa una inicialización o puesta a cero completa, de texto y zona de variables simultáneamente.



Una vez detenido el programa puede reanudarse su ejecución por medio de la sentencia **CONT.**

DETENCION DE LA EJECUCION DE UN PROGRAMA

Ya que sabemos poner en funcionamiento un programa BASIC, parece normal que dispongamos de una forma de detenerlo. El sistema más inmediato de forzar una detención en la ejecución es por medio de la pulsación de la tecla **RUN STOP.** Esta tecla produce la interrupción inmediata acompañada del mensaje **BREAK IN...** (detenido en...) que indica el número de instrucción en que se produjo la interrupción.



La sentencia **CLEAR** *no afecta en absoluto a la zona de texto BASIC.*

Una vez detenido el programa puede reanudarse su ejecución desde el mismo punto por medio de la sentencia BASIC **CONT** (continuar), o bien optar por ejecutar cualquier otro comando directo como por ejemplo un **GOTO** a un número de línea determinado para continuar.

Como advertencia diremos que conviene rehuir el empleo para la nueva puesta en marcha del programa de la palabra clave **RUN** especificando o no algún número de línea, ya que esto implicaría el borrado del área de variables, con consecuencias impredecibles en. la continuación de la ejecución.

Un efecto similar al de la pulsación de la tecla **RUN STOP** puede conseguirse por medio de la sentencia BASIC **STOP**. Esta puede incluirse en cualquier parte del programa, produciendo el conocido mensaje **BREAK IN....** **STOP** tiene como ventaja sobre **RUN STOP** el saberse de antemano el punto exacto en que se detendrá el programa lo que, en la mayoría de las ocasiones es absolutamente impredecible por el otro sistema.

STOP puede permitirnos averiguar, después de la detención, el contenido de determinadas variables, cosa ésta de gran valor a la hora de depurar un posible mal funcionamiento del programa; reanudándose igualmente la ejecución por medio de **CONT.**

Existe además una forma de marcar el final del programa BASIC de forma que no es posible

STOP *nos permite averiguar el contenido de determinadas variables en el momento de una interrupción.*



continuar su ejecución por medio de **CONT.** La sentencia que produce este efecto es **END** (fin).

La palabra clave **END** es especialmente útil cuando se incluye en alguna línea del programa que debe ser la última en ejecutarse aunque no ocupe el último lugar en la secuencia ascendente de los números de instrucción. Además, tiene como ventaja adicional que no produce el mensaje de error **BREAK IN ...** sino que cede el control directamente al intérprete del BASIC, en comando directo, con el mensaje **READY.**

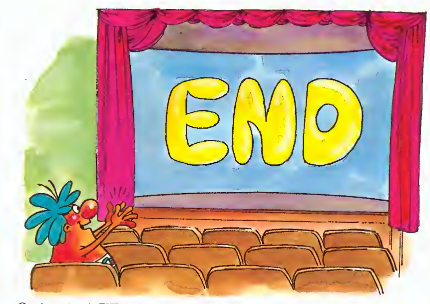
Las sentencias que hemos comentado se adaptan bastante a nuestras necesidades, sin embargo, es preciso en ocasiones incluir una pausa en el programa BASIC que permita su posterior reanudación con la pulsación de una tecla, o transcurrido determinado tiempo.

Para el primer caso podemos optar por dos soluciones diferentes: la inclusión de una sentencia **INPUT** de cualquier variable, o la de una **GET,** como vemos en el ejemplo:

```
50 INPUT X$

80 GET X$:IF X$="" THEN 80
```

A la primera de ellas podemos contestar con la pulsación de **RETURN** solamente, ya que es la única forma de salir de un **INPUT**; teniendo, además, como inconveniente el encendido del cursor de programa y la impresión en pantalla del mensaje de petición de datos del Sistema**?.**



Con la sentencia **END** *marcamos el final de un programa, sin posibilidad de continuación por* **CONT.**

En el segundo caso obtenemos una detención real del programa hasta que el usuario pulse cualquier tecla, diferente de **SHIFT, CTRL, COMMODORE** o **RESTORE** que no tienen significado por sí mismas, sin los problemas expuestos anteriormente.

Por último, la solución al problema de una pausa temporizada puede ser la inclusión de un bucle **FOR NEXT** en vacío, o el empleo del reloj del Sistema. Veamos un ejemplo del primer caso:

```
30 FOR I=0 TO 999:NEXT
```

El empleo del reloj del Sistema es algo más complicado pero puede permitirnos ajustar a una medida de tiempo convencional la duración de la pausa. El reloj del Sistema se actualiza cada cincuentavo de segundo (dos centésimas) sumando uno a la variable del Sistema **TI,** cuyo valor se incrementa sucesivamente desde que se conecta el ordenador. El siguiente programa produce una pausa de dos segundos:

```
70 X=TI+100:IF X<TI THEN 70
```

El Sistema dispone también de la variable alfanumérica **TI$** de seis caracteres, conteniendo la hora, minutos y segundos. Esta variable se pone a cero al conectar el ordenador de forma que si preguntamos su valor en un determinado momento nos dirá el tiempo transcurrido desde la conexión.

Podemos también «poner en hora» este reloj, por **INPUT** o por asignación directa. Para ponerlo en las cinco y cuarto de la tarde, por ejemplo, escribiremos:

```
TI$="171500"
```

Esta variable se actualiza de forma automática y podemos utilizarla sin restricciones en nuestros propios programas.

NEW *implica una puesta a cero de texto BASIC y zona de variables, simultáneamente*

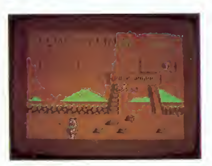


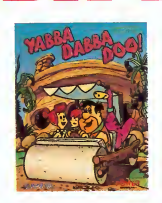
YABBA DABBA DOO

Desde que la vida apareció en la Tierra, la evolución ha ido propiciando nuevas formas de existencia, desde las primeras bacterias, pasando por los protozoos, reptiles, mamíferos, aves, hasta llegar a su culmen: el hombre, tal y como lo conocemos.

Pero lo que más nos interesa en estos momentos es hallar nuestras propias raices, provenientes, como todos sabemos, del mono. Aunque nadie se haya fijado, antes del mono, hubo en los lejanos tiempos en que los dinosaurios poblaban la Tierra, una simpática familia con unos adelantos técnicos muy grandes; estos conocían ya la rueda, el coche, la gasolina, el cine, el bingo, las boleras... lo que pasa es que siempre lo guardaron muy en secreto, eran... los Picapiedra, con Pedro Picapiedra y Pablo Mármol como personajes notables de esta famosa tribu. Sus







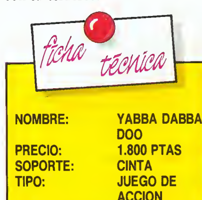
respectivas, se llamaban Wilma y Cuchi-Cuchi, y su simpática mascota Dino, sus hijos, terribles pequeñines de la edad de piedra atendían por Bam-Bam y Pebbles.

Pero dejemos el árbol genealógico de esta simpática familia y vayamos a centrarnos en los personajes que verdaderamente nos interesan en esta ocasión, el matrimonio Picapiedra, compuesto por Pedro y Wilma. Remontándonos a los tiempos en que eran únicamente novios (algo que por entonces se llevaba mucho), Pedro tenía el más firme deseo de hacer de Wilma su mujer (como después ocurriría), pero ésta, chica muy responsable, exigía a Pedro que para hacer efectiva esta boda, como dote debería presentar una casa, que haría de nidito de amor en el futuro.

Puestos ya en antecedentes, tendremos que ayudar a Pedro a construir su vivienda, y para ello deberemos primero buscar a Wilma, que le ayudará en su empeño. A partir de los desolados terrenos prehistóricos esta casa será el principio de la gran urbanización de Rocadura. Cuando hayamos encontrado a Wilma, guiaremos a Pedro hasta la cantera, de donde deberá ir recogiendo piedras para la construcción.

Pero no todo será tan fácil, ya que numerosos animales que vagarán por los alrededores molestarán muy asíduamente, en especial Dino, la traviesa mascota de los Picapiedra. Para ir haciendo el trabajo más rápido podremos valernos del Troncomóvil, si logramos encontrarlo y nos acordamos de ponerle gasolina de vez en cuando.

Así que ya sabes, ayuda a Pedro y cambia el curso de la historia con tu esfuerzo.





SLURPY

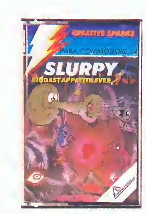
Nos hallamos en las recónditas cuevas subterráneas del planeta Gluton, lugar donde habita un simpático, pero al mismo tiempo extrañísimo ser.

Nuestro amiguito únicamente tienen un problema, pero gravísimo como podréis observar: nació con una increible trompa, a guisa de boca, y un hambre insaciable que le acosa día y noche desde entonces. De este modo, se independizó de su familia y se fue a vivir su vida, desarrollando una poderosa fuerza en la trompa, que le permitía calmar su voracidad, pudiendo absorber a todo ser viviente que caía dentro de su radio de acción.

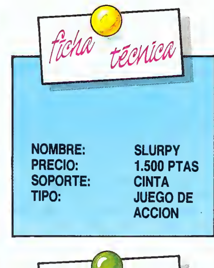






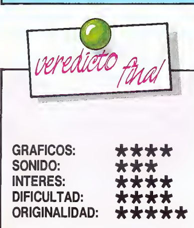


Por el contrario, el cuerpo de Slurpy es muy débil, y cualquier contacto con el resto de animales le conllevará la muerte, por ello debemos ayudarle en su defensa, y maniobrar certeramente su trompa para absorber a sus enemigos antes de que le toquen.



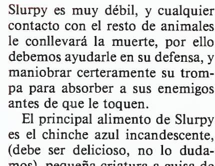
ficha técnica

NOMBRE:	SLURPY
PRECIO:	1.500 PTAS
SOPORTE:	CINTA
TIPO:	JUEGO DE ACCION



veredicto final

GRAFICOS:	★★★★
SONIDO:	★★★
INTERES:	★★★★
DIFICULTAD:	★★★★
ORIGINALIDAD:	★★★★★



El principal alimento de Slurpy es el chinche azul incandescente, (debe ser delicioso, no lo dudamos), pequeña criatura a guisa de bolita reluciente, con las que Slurpy debe acabar completamente para llegar a otra cueva (aviso: son numerosísimos). Estos chinches, debido a determinadas condiciones químicas de la cueva, cambian de color, de azul a rojo y a color oro; si Slurpy traga un rojo morirá, por lo que debe escupirlo rápidamente si ha incurrido en ese error. Si logra absorber uno dorado, éste le dará fuerzas suplementarias.

En la parte inferior de cada cueva hay una docena de huevos que eclosionarán al ser tocados por un chinche rojo o cualquier otra criatura del interior de las cuevas, pero serán reemplazados por posteriores puestas de los mismos animalejos.

Cuidado con la gravedad del planeta, pues de vez en cuando provoca terribles desprendimientos que pueden acabar con Slurpy. Al mismo tiempo, la situación se irá haciendo progresivamente más difícil a medida que avancemos por las sucesivas cuevas.

Sobre todo, recuerda no ser excesivamente glotón, puesto que no podrás tragar varios animales de una sola «absorbida», aunque también gozarás de una buena posibilidad: la de moverte mientras comes; ¡te será muy útil! De modo que ¡ánimo y a por ellos!

CRAZY GOLF

Para que no se comente que no nos ocupamos de los hermanos más pequeños del Commodore-64, vamos a abrir unos nuevos Softmodore para el 16, con el fin de que los usuarios de esta pequeña maravilla se sientan incluidos también en Tu Micro Commodore.

Con el auge del golf en nuestro país, a causa de los éxitos por toda Europa y EE.UU. de la «escuadra española», representada por Ballesteros, Piñero, Cañizares..., este entretenido deporte ha llegado también a las pantallas del Commodore-16; para que podamos ser unos campeones de este deporte sentados tranquilamente en un sillón, delante del televisor.

A pesar de la no muy potente memoria de este ordenador, este juego esta muy conseguido y presenta un gran grado de entretenimiento para el usuario.

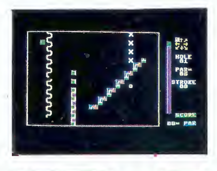
Como todos sabemos, el golf consiste en realizar un recorrido a unos determinados hoyos previamente determinados, donde partiendo del punto que se ha dado como salida de ese hoyo, habrá que lograr introducir la pelota en él, ¡pero alto! No tenemos toda la vida para ello, puesto que cada hoyo se halla estandarizado en un determinado número de golpes.

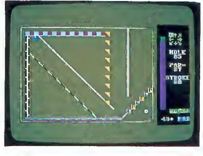


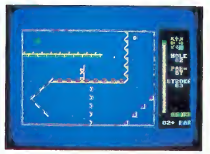
De este modo, el objetivo será realizarlo en el menor número de golpes posible.

El estandard de golpes que se cree son suficientes para introducir la pelota en el hoyo, es el Par; cualquier golpe por encima de éste, sera Bogey, doble Bogey...; si por el contrario, se realiza el hoyo en menos golpes de los indicados habremos conseguido un Birdie, si ha sido en un golpe menos, y un Eagle si lo hemos realizado en dos, o sea dos bajo par. Con estas nociones básicas ya deberíamos estar preparados para empezar a jugar a este apasionante deporte, sin ninguna dificultad mayor que la de nuestra pericia y precisión en los golpes.

En el Crazy Golf será un poco más difícil, ya que nos encontraremos en nuestro camino con multitud de obstáculos, y los rebotes de la bola serán en cierto modo imprevisibles. Tendremos que ajustar antes de cada golpe la dirección que queremos lleve nuestra pelota, así como la fuerza, indicada por una barra. Cualquier cálculo erróneo hará que perdamos gran parte de nuestro handicap para alcanzar un buen número de golpes, y que acabemos el recorrido en 20 o 30 sobre par (lo decimos por experiencia), pero no hay que desanimarse, y empezar a jugar otra vez, con el tiempo nos convertiremos en verdaderos campeones.









NOMBRE:	CRAZY GOLF
PRECIO:	1.500 PTAS
SOPORTE:	CINTA:
TIPO:	DEPORTES



GRAFICOS:	*
SONIDO:	**
INTERES:	****
DIFICULTAD:	****
ORIGINALIDAD:	**

¡FUERA ERRORES!

C-Byte tiene el honor de presentar, en exclusiva nacional para los lectores de TU MICRO COMMODORE, el revolucionario sistema de introducción de programas FUERA ERRORES. Este nos permitirá introducir, sin temor alguno al esfuerzo inútil, cualquier listado por largo y complicado que parezca.

Para adoptar los listados publicados bajo este sistema, deberemos seguir las siguientes normas:

1) Es fundamental transcribir EXACTAMENTE el listado reproducido, incluyendo todos sus espacios, aunque se trate de separaciones entre número de instrucción y línea de instrucción.

2) Todas las líneas finalizarán con un número de tres dígitos, encerrado entre guiones, que NO deberá ser introducido, puesto que no forma parte del programa, sino que tiene la finalidad de hacer funcionar el sistema FUERA ERRORES, según veremos más adelante. Para evitar equivocaciones, dicha cifra entre guiones se sitúa en el margen derecho del final de la línea BASIC a la cual corresponde, a una distancia prudencial del mismo.

3) Para facilitar la introducción de símbolos difícilmente interpretables, se procede a la siguiente representación en los listados.

— Las letras aparecidas entre menor y mayor deberán ser introducidas con pulsación simultánea de la tecla COMMODORE y la letra representada. Ej.: <M> =COMMODORE M.

— Las letras aparecidas entre barras verticales deberán ser introducidas como pulsación simultánea de la tecla SHIFT y la letra representada. Ej.: ¦ K ¦ = SHIFT K.

— Entre corchetes simples se representarán los símbolos que se obtienen por pulsación directa de la tecla, aunque lógicamente, este caso sólo se dará para indicar las sucesiones de más de una letra. Así por ejemplo, la introducción de 5 asteriscos se representaría por [5*].

— Para la repetición de símbolos obtenidos mediante las teclas COMMODORE o SHIFT, se seguirá una combinación de las tres normas anteriormente citadas. Así por ejemplo, la introducción de 10 símbolos COMMODORE H, se representaría por [<10 H>].

— Para evitar confusiones, cuando se utilice el sistema de representación de sucesiones de carácter, y éste sea un espacio, se utilizará la abreviatura ESP. [15 ESP] = 15 espacios.

— Los carácteres de control, tales como desplazamientos del cursor, colores, estados de reversa y funciones, se símbolizarán por una abreviatura de tres letras (dos más un espacio en el caso de las funciones) encerrada entre llaves, tal como se señala en la tabla adjunta.

Para introducir cualquier listado por el sistema FUERA ERRORES, deberemos entrar previamente y ejecutar el listado BASIC que aparece en la página siguiente por lo cual es recomendable conservar una copia grabada del mismo, para sucesivas ocasiones. Una vez introducido este listado, ya sea por el teclado, o a través de cinta o disco, debemos ejecutarlo con **RUN.** Instantes más tarde aparecerá en la pantalla el mensaje **FUERA ERRORES!** y el cursor libre para la entrada de programas, con el tradicional **READY.** por encima, lo cual indicará la activación del sistema de depuración de errores.

En virtud al **NEW** que finaliza la línea 20 del programa ¡FUERA ERRORES!, este habrá desaparecido de la memoria, y seremos libres para introducir cualquiera de los programas listados en la sección **TECLA A TECLA** de cualquier número de nuestra revista, o incluso aquellos que apareciendo en otras secciones se acojan a este sistema. Así pues, si el programa ¡FUERA ERRORES! ha desaparecido de la memoria, ¿qué hemos conseguido ejecutándolo? Bien, la respuesta se llama informáticamente **INTERRUPCION:** se trata de una técnica de programación en código máquina que permite que el ordenador efectúe prácticamente dos trabajos a un tiempo, o más correctamente, que ejecute determinada tarea de forma automática, sin necesidad de que le prestemos una atención constante, de forma similar al proceso de respiración en un humano.

Efectivamente, aunque el soporte BASIC ha desaparecido de la memoria, antes de «marcharse» ha dejado funcionando en modo interrupción la pequeña rutina en código máquina que se hallaba en sus **DATAS.** Para comprobarlo pulsa **RETURN**; observarás algo muy extraño: tu ordenador no se comporta normalmente, no sólo desciende una línea el cursor, sino que además hace aparecer un número en la esquina superior izquierda de la pantalla. A continuación veremos como emplearlo.

Cada vez que pulsemos **RETURN**, aparecerá un número en la mencionada zona de la pantalla, y este corresponderá con la instrucción que hayamos introducido. Esto forma parte del sistema de FUERA ERRORES. Cuando introduzcamos cualquier línea de un listado de este tipo, deberemos fijarnos en el número que aparece al pulsar el **RETURN** de fin de línea; si este coincide con el que aprece en el listado al final de la línea, ésta habrá sido introducida correctamente, en caso contrario existe algún error de teclado que debemos modificar. Para modificar una instrucción errónea, no tenemos ni tan siquiera que volver a teclearla si no queremos, bastará sencillamente con modificar el carácter o caracteres erróneos como siempre hacemos, hasta que nos coincida el número de verificación que se presentará al pulsar **RETURN.**

Así pues, el sistema FUERA ERRORES se compone de dos partes: una codificación especial de los listados que facilitan su introducción, evitando los errores al confundir los caracteres gráficos, de control, etc... y un sistema de verificación de líneas que nos advierte en el preciso instante de introducir una de estas, que está mal tecleada.

Ahora bien, el empleo de estos dos sistemas no quiere decir que nos encontremos ante un BASIC diferente al de COMMODORE 64. Este no ha cambiado, hace exactamente las mismas cosas de siempre; simplemente hemos cambiado la forma de hacer los listados. En cuanto al misterioso número que aparece en la esquina de la pantalla no es más que

una simple suma de comprobación, lo que se conoce técnicamente como un CHECKSUM. La rutina en código máquina de interrupción suma los valores de los caracteres que entramos en cada línea que introducimos, aplicándoles un módulo 256, es decir, volviendo a cero cada vez que su suma parcial supera el 255, de este modo se crea un número entre 0 y 255 dependiente directamente de los caracteres introducidos. Así pues, es prácticamente imposible que una sucesión de errores den por casualidad ese número, mientras que siempre que la línea esté correctamente introducida obtendremos el mismo código que nosotros hallamos por igual sistema y añadimos al final de cada línea al realizar el listado del programa.

Por tanto, el misterioso número no es tampoco una modificación del BASIC de COMMODORE, sino simplemente un pequeño truco para la comprobación de que las líneas han sido bien introducidas. Propiamente no nos evita cometer errores de tecleado, sino que simplemente nos advierte inmediatamente en que línea los hemos introducido.

Para desactivar el sistema sólo deberemos pulsar RUN/STOP RESTORE, y si por cualquier motivo nos interesara reactivarlo, podríamos ejecutar SYS 822, siempre y cuando se encuentre el código máquina en la memoria, lógicamente.

¡ADVERTENCIA! Puesto que el código máquina se encuentra ubicado en el buffer del casete, es imprescindible desactivarlo (RUN/STOP RESTORE) antes de realizar cualquier operación con dicho periférico.

UTILIZACION DEL CASETE CON FUERA ERRORES

Si queremos introducir parte de un programa, para continuar posteriormente el trabajo emprendido, sin perder por supuesto la enorme ventaja del FUERA ERRORES deberemos proceder de la siguiente forma:

1. Desactivar el sistema FUERA ERRORES mediante la pulsación de RUN/STOP + RESTORE.
2. Trasladar el código máquina desde su ubicación en el buffer del casete hasta otro punto desocupado en la memoria RAM (por ejemplo 49152). Para ello sólo hemos de saber que ocupa 114 bytes desde 822 (inclusive). Una sencilla rutina que efectúe el trabajo de reubicación puede ser: FORI=0TO113:POKE49152+I,PEEK(822+I):NEXT.
3. Realizar la operación correspondiente con el casete; ya sea grabar una copia de seguridad de lo introducido en la memoria hasta el momento, ya sea cargar del casete una parte ya grabada del programa.
4. Restablecer FUERA ERRORES a su punto original. Siguiendo el caso de la línea de ejemplo anterior: FORI=0TO113:POKE822+I,PEEK(49152+I):NEXT.
5. Reactivar el sistema FUERA ERRORES mediante SYS 822 (esta vez no aparecerá mensaje alguno, aunque al pulsar **RETURN** comprobaremos que aparecen las cifras de control en la esquina superior izquierda). Lógicamente, los pasos 4 y 5 no son necesarios si después de una grabación se va a apagar el ordenador, y sólo se llevarán a cabo después de las cargas, o si tras la grabación de una copia de seguridad del programa, deseamos continuar introduciéndolo acto seguido.

TABLA DE INTERPRETACION DE CODIGOS DE CONTROL

ABR	SIFNIFICADO	OBTENCION	ABR	SIFNIFICADO	OBTENCION
HOM	HOME	CLR/HOME	F8	FUNCION 8	SHIFT F7
CLR	CLEAR + HOME	SHIFT CLR/HOME	BLK	BLACK (NEGRO)	CTRL 1
ABJ	CURSOR ABAJO	CRSR VERTICAL	WHT	WHITE (BLANCO)	CTRL 2
ARB	CURSOR ARRIBA	SHIFT CRSR VERTICAL	RED	RED (ROJO)	CTRL 3
			CYN	CYAN (CIAN)	CTRL 4
DCH	CURSOR DERECHA	CRSR HORIZONTAL	PUR	PURPPLE (PURPURA)	CTRL 5
IZQ	CURSOR IZQUIERDA	SHIFT CSRS HORIZONTAL	GRN	GREEN (VERDE)	CTRL 6
			BLU	BLUE (AZUL)	CTRL 7
RON	REVERSE ON	CTRL 9	YEL	YELLOW (AMARILLO)	CTRL 8
ROF	REVERSE OFF	CTRL 0	NRJ	NARANJA	COMMODORE 1
F1	FUNCION 1	F1	MRN	MARRON	COMMODORE 2
F2	FUNCION 2	SHIFT F1	RCL	ROJO CLARO	COMMODORE 3
F3	FUNCION 3	F3	GR1	GRIS 1	COMMODORE 4
F4	FUNCION 4	SHIFT F3	GR2	GRIS 2	COMMODORE 5
F5	FUNCION 5	F5	VCL	VERDE CLARO	COMMODORE 6
F6	FUNCION 6	SHIFT F5	ACL	AZUL CLARO	COMMODORE 7
F7	FUNCION 7	F7	GR3	GRIS 3	COMMODORE 8

```
10 FORI=822TO935:READA:C=C+A:POKEI,A:NEXT
20 IFC<>15254THENPRINTCHR$(147)"ATENCION!, HAY UN ERROR EN LOS DATOS":END
30 PRINTCHR$(147)TAB(213)"FUERA ERRORES!":SYS822:NEW
100 DATA 169,3,141,37,3,169,69,141,36,3,169,0,133,254,96,32,87,241,133,251
110 DATA 134,252,132,253,8,201,13,240,13,24,101,254,133,254,165,251,166,252
120 DATA 164,253,40,96,169,13,32,210,255,165,214,141,176,3,206,176,3,169,0
130 DATA 133,216,169,18,32,210,255,169,19,32,210,255,169,45,32,210,255,166
140 DATA 254,224,100,176,5,169,48,32,210,255,224,10,176,5,169,48,32,210,255
150 DATA 169,0,133,254,32,205,189,169,45,32,210,255,173,176,3,133,214,76,88,3
```

PANTAMON

Crear pantallas espectaculares es algo sencillo de hacer en nuestro Commodore. Lo realmente difícil es producir las sentencias PRINT que nos permitan introducir esas pantallas en nuestros programas. Para resolver este problema, aquí os traemos a Pantamon.

PANTAMON es un programa de utilidad que puede ahorrarte bastantes quebraderos de cabeza y horas de trabajo cuando quieras producir, por ejemplo, pantallas con título de presentación para tus programas o paisajes para tus juegos.

El uso de PANTAMON es extremadamente sencillo. Teclea el listado adjunto, y haz una copia en cinta o disco, pues el programa se autodestruirá (no en cinco segundos, como en las películas de espías, sino al terminar su función).

Al hacer RUN, aparecerá el rótulo de presentación que permanecerá estático hasta que pulses una tecla. Este rótulo ha sido producido con una versión inicial de este mismo programa; incluso ahora, con las líneas **PRINT** ya corregidas, podremos comprobar lo pesado que resulta teclear esta parte. Si además hubiésemos tenido que producirla «a mano»...



Para ahorrar tiempo, los que lo deseen pueden saltarse esta parte (ellos se perderán la hermosa pantalla, que además de presentación sirve de ejemplo). Eso si, no cambiéis los números de línea en el programa; saltad la parte de PRINT sin modificar nada.

LA PANTALLA DE TRABAJO SE TRATARA IGUAL QUE LA PANTALLA DEL COMMODORE

Una vez hecho esto, entraremos en la pantalla de trabajo. Disponemos de un cursor, el cual nos indicará el punto de pantalla en la que vamos a imprimir. Para dibujar, simplemente nos movemos y pulsamos la tecla cuyo gráfico deseemos obtener.

Todas las opciones del editor de pantalla del Commodore están disponibles aquí: podemos usar caracteres en inverso, cambiar el color, mover el cursor, etc. tal como lo haríamos normalmente.



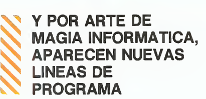
Un detalle: Al usar la inserción o el borrado de caracteres (insert/delete) ocurre frecuentemente que la línea inmediatamente debajo de aquella en la que estamos sufre también la inserción o el borrado.

Esto es debido a la forma de operar del editor de pantalla, que trabaja de dos en dos líneas (para permitir la entrada de sentencias Basic de hasta 80 caracteres). Para solucionarlos, nos movemos a esa línea e insertamos o borramos lo necesario hasta que quede correcta.

Además, si intentamos pasar de la línea inferior de la pantalla se producirá un scroll de toda la pantalla hacia arriba, lo cual puede estropearnos nuestro hermoso (!) trabajo, por lo que hemos de tener cuidado. Claro que, a veces podemos hacerlo intencionadamente para desplazar el dibujo.

En fin, para recordar todos estos aspectos, nos basta recordar que estamos trabajando con el commodore exactamente igual que si dibujásemos sobre su pantalla en modo directo.

Con las teclas de función F1 y F3 podremos cambiar los colores del marco y el fondo de la pantalla. Para cambiar el color de los caracteres, haremos lo de siempre: Pulsando CTRL (o la tecla Commodore accionaremos una tecla numérica para obtener el color correspondiente.



Y POR ARTE DE MAGIA INFORMATICA, APARECEN NUEVAS LINEAS DE PROGRAMA

Una vez que hayamos obtenido el resultado deseado, pulsaremos la tecla F5. El mensaje ESPERE aparecerá en grandes letras sobre la pantalla, mientras el ordenador efectúa todos los cálculos necesarios (que trabaje él, que para eso lo tenemos) que de otro modo deberíamos hacer nosotros.

Una vez terminado el proceso, PANTAMON desaparecerá de la memoria. Si hacemos un **LIST**, veremos que han aparecido unas líneas (de 5.000 en adelante) que nosotros no habíamos tecleado, y que contienen las sentencias suficientes para producir la pantalla inicial.

Esas líneas están a nuestra disposición, para incorporarlas a nuestros programas. Si la numeración producida no es la adecuada para nosotros, simplemente deberemos cambiarla (poco trabajo después de todo, ¿no?).

Unicamente hemos de corregir un detalle para evitar que la pantalla haga un scroll cuando se imprima la última línea, lo cual podría arruinar el resultado: listamos la ULTIMA línea que el programa haya generado, eliminamos el último caracter de dicha línea y lo sustituimos por un HOME. Con esto, el dibujo permanecerá inalterable en pantalla.

Ahora es el turno de sacarle partido a esta nueva utilidad. Sólo nuestra imaginación (y no nuestra pereza, como hasta ahora) será el turno para la espectacularidad de nuestras pantallas.

¿Y cómo se han generado esas líneas? Magicamente, desde luego, pero con la magia de la informática.

Si observáis atentamente como funciona el programa mientras esperamos, podréis observar que lo que hace es imprimir en la pantalla un número, un símbolo «?»,

comillas y un montón de datos detrás.

Estos datos son generados por el programa principal, y es exactamente lo que deberíamos introducir en una sentencia PRINT para producir una línea de pantalla.

A continuación, llenamos el buffer de teclado con un HOME y varios RETURN, mediante pokes a las direcciones 631-640, e informamos que el buffer esta lleno con **POKE 198,9.** Si ahora paramos el programa, el ordenador imprimirá los caracteres que hemos introducido en el buffer, creyendo que hemos pulsado las teclas correspondientes.

Así, vuelve a HOME y, al encontrar un RETURN, introduce en memoria la línea de programa que habíamos impreso. Si deseais una información más detallada, podeis encontarla en la CHISTERA del número 4 (generador de DATAS).

Animo con ese teclado, y veamos el efecto del programa en las pantallas de presentación.

```
1REM ********************************          *
                                  *                   -169-
2 REM * PANTAMON 1986 SINDIN SOFT     *               -225-
3 REM *                               *             *******
*************************                            -171-
4 PRINT"{CLR}{ROF}";:POKE650,128:GOSUB58:A=32:I=.
                                                      -130-
5 FORI=.TO23:PRINT"[40 ESP]";:NEXT                    -183-
6 PRINT"{HOM}";:GOTO78                                -087-
7 I=PEEK(209)+256*PEEK(210)+PEEK(211):RETURN
                                                      -056-
8 PRINT"<+>{IZQ}";                                    -167-
9 GETA$:IFA$=""THEN9                                  -181-
10 IFA$="{RON}"THENRE=1                               -060-
11 IFA$="{ROF}"THENRE=.                               -186-
12 IFA$="{F1 }"THENPOKE53280,(PEEK(53280)AND15)+1:GOT
08                                                    -018-
13 IFA$="{F3 }"THENPOKE53281,(PEEK(53281)AND15)+1:GOT
08                                                    -022-
14 IFA$="{F5 }"THEN18                                 -025-
15 GOSUB7:POKEI,A:POKEI+54272,B:IFRETHENPRINT"{RON}";
                                                      -088-
16 PRINTA$;:GOSUB7:A=PEEK(I):B=PEEK(I+54272)
                                                      -195-
17 GOTO8                                              -249-
18 GOSUB7:POKEI,A:POKEI+54272,8                       -174-
19 POKEV+21,63:POKE53280,.:POKE53281,.                -189-
20 A=49152:B=51000:FORI=.TO999                        -215-
21 POKEA+I,PEEK(1024+I):POKEB+I,PEEK(55296+I)
                                                      -132-
22 NEXT                                               -195-
23 PRINT"{CLR}{GRN}{ROF}";                            -212-
24 FORI=828TO900:POKEI,.:NEXT:POKE828,1               -190-
25 RE=.:CA=.:PRINT"{CLR}{GRN}"5000+10*PEEK(828)"{IZQ}
?";CHR$(34);:POKE828,PEEK(828)+1                      -251-
26 C=.:FORI=.TO39:A=PEEK(49152+I+40*PEEK(830))
                                                      -139-
27 B=AAND127:IFB<32THENB=BOR64:GOTO30                 -051-
28 IFB>63ANDB<96THENB=B+32:GOTO30                     -024-
29 IFB>95THENB=B+64                                   -141-
30 IF(AAND128)<>.THENIFRE=.THENRE=1:PRINT"{RON}";:C=C
+1                                                    -095-
31 IFA<128THENIFRE=1THENRE=.:PRINT"{ROF}";:C=C+I
                                                      -080-
32 CO=PEEK(51000+I+40*PEEK(830))AND15:IFCO=CATHEN50
                                                      -231-
33 CA=CO:IFCO=.THENPRINT"{BLK}";                      -106-
34 IFCO=1THENPRINT"{WHT}";:GOTO49                     -054-
35 IFCO=2THENPRINT"{RED}";:GOTO49                     -079-
36 IFCO=3THENPRINT"{CYN}";:GOTO49                     -212-
37 IFCO=4THENPRINT"{PUR}";:GOTO49                     -211-
38 IFCO=5THENPRINT"{GRN}";:GOTO49                     -087-
39 IFCO=6THENPRINT"{BLU}";:GOTO49                     -090-
40 IFCO=7THENPRINT"{YEL}";:GOTO49                     -210-
41 IFCO=8THENPRINT"{NRJ}";:GOTO49                     -183-
42 IFCO=9THENPRINT"{MRN}";:GOTO49                     -205-
43 IFCO=10THENPRINT"{RCL}";:GOTO49                    -247-
44 IFCO=11THENPRINT"{GR1}";:GOTO49                    -250-
45 IFCO=12THENPRINT"{GR2}";:GOTO49                    -253-
46 IFCO=13THENPRINT"{VCL}";:GOTO49                    -000-
47 IFCO=14THENPRINT"{ACL}";:GOTO49                    -003-
48 IFCO=15THENPRINT"{GR3}";                           -078-
49 C=C+1                                              -172-
50 PRINTCHR$(B);:C=C+1                                -058-
51 IFC<65THEN54                                       -151-
52 PRINTCHR$(34)";":PRINT5000+10*PEEK(828)"?"CHR$(34)
;                                                     -029-
53 POKE828,PEEK(828)+1:C=.                            -225-
54 NEXT:PRINTCHR$(34)";":POKE198,9:POKE631,19:FORI=63
2TO640:POKEI,13:NEXT                                  -010-
55 POKE830,PEEK(830)+1:IFPEEK(830)=26THEN57           -007-
56 PRINT"GOTO 25":END                                 -045-
57 POKE198,.:PRINT"{CLR}{GRN}";:POKE828,.:GOTO116
                                                      -030-
58 DATA0,0,0,63,255,128,63,255,128,63,255,128,63,255,
128,63                                                -180-
59 DATA0,0,63,0,0,63,0,0,63,0,0,63,248,0,63,248
                                                      -053-
60 DATA0,63,248,0,63,248,0,63,0,0,63,0,0,63,0,0
                                                      -045-
61 DATA63,0,0,63,255,128,63,255,128,63,255,128,63,255
,128,0                                                -174-
62 DATA0,0,0,63,255,128,63,255,128,63,255,128,63,255,
128,63                                                -175-
63 DATA0,0,63,0,0,63,0,0,63,0,0,63,255,128,63,255
                                                      -151-
64 DATA128,63,255,128,63,255,128,0,31,128,0,31,128,0,
31,128                                                -160-
65 DATA0,31,128,63,255,128,63,255,128,63,255,128,63,2
55,128,0                                              -024-
66 DATA0,0,0,63,255,0,63,255,0,63,255,0,63,255,0,63
                                                      -007-
67 DATA15,0,63,15,0,63,15,0,63,15,0,63,255,0,63,255
                                                      -008-
68 DATA0,63,255,0,63,255,0,63,0,0,63,0,0,63,0,0
                                                      -049-
69 DATA63,0,0,63,0,0,63,0,0,63,0,0,63,0,0,0          -090-
70 DATA0,0,0,63,255,128,63,255,128,63,255,128,63,255,
128,63                                                -174-
71 DATA15,128,63,15,128,63,15,128,63,15,128,63,255,12
8,63,255                                              -026-
72 DATA128,63,255,128,63,255,128,63,31,0,63,31,0,63,1
5,128                                                 -118-
73 DATA63,15,128,63,7,192,63,7,192,63,3,224,63,3,224,
0                                                     -178-
74 FORI=16128TO16383:READA:POKEI,A:NEXT:POKE2040,252:
POKE2041,253:POKE2042,254                             -179-
75 POKE2043,252:POKE2044,255:POKE2045,252:V=53248
                                                      -016-
76 FORI=.TO10STEP2:POKEV+I,90+15*I:POKEV+1+I,120:NEXT
                                                      -045-
77 FORI=V+39TOI+6:POKEI,5:NEXT:RETURN                 -219-
78 PRINT"{CLR}{BLK}";:POKE53280,.:POKE53281,.
                                                      -073-
79 PRINT" {ACL} {WHT}[6 ESP]SINDIN[3 ESP]SOFT[3 ESP]P
RESENTA[4 ESP]{ACL}[4 ESP]";                          -248-
80 PRINT"{ACL}[40 ESP]";                              -046-
81 PRINT"{ACL}[40 ESP]";                              -047-
82 PRINT"{ACL}[6 ESP]{RON}{BLU}<BLU>[2 ESP]!N!{ROF}{A
CL} {BLU} {ACL}[2 ESP]{RON}{BLU}<BLU> !N!{ROF}{ACL} {
BLU} {RON}<M>{ROF} {ACL} {RON}{BLU}<M>{ROF} {RON}!M![
2 ESP]!ESP!{ROF}{ACL} {BLU} {ACL} {RON}{BLU}!M! !M!{R
OF}{ACL}[4 ESP]";                                     -043-
83 PRINT"[2 ESP]";                                    -215-
84 PRINT"{ACL}[5 ESP]{RON}{BLU}<BLU>[2 ESP]!3 N!{ROF}
 {ACL} {RON}{BLU}<BLU> !3 N!{ROF} {RON}<2 M>{ROF}{ACL
} {RON}{BLU}<M>{ROF} !ROF!{RON}!M![2 ESP]!ESP!{ROF} {
RON}!3 M! !M!{ROF}{ACL}[5 ESP]";                      -225-
85 PRINT"{ACL}[2 ESP]{PUR}[3 ESP]{RON}[3 ESP]{BLU}!2
N!{ROF}<ROF> {RON}<RON>{PUR}[2 ESP]{BLU}!2 N! {ROF} {
RON}{PUR} {BLU}<2 M>{PUR} {ROF}{ACL} {BLU} {RON}!M!{P
UR}[3 ESP]{ROF}{BLU} {RON} !2 M!{PUR}[2 ESP]{BLU}!BLU
!{ROF}{ACL}[4 ESP]";                                  -155-
86 PRINT"";                                           -154-
87 PRINT"{PUR}[5 ESP]{RON} {BLU}!N! {PUR} {ROF}{BLU}<
BLU>{ACL} {BLU} {RON}{PUR} {BLU}[2 ESP]{PUR} {BLU}[2
ESP]{ROF} {RON}{PUR}[2 ESP]{BLU}<M>{PUR} {ROF}{ACL} {
BLU} {RON}[2 ESP]{PUR} {ROF}{ACL} {BLU} {RON}[2 ESP]{
PUR} {BLU}[2 ESP]{PUR} ";                             -220-
88 PRINT"{ROF}{ACL}[4 ESP]";                          -072-
89 PRINT"{ACL}[3 ESP]{PUR} {ACL} {RON}{PUR}[3 ESP]{RO
F}{ACL}[2 ESP]{BLU} {ACL} {RON}{PUR}[4 ESP]{BLU}[2 ES
P]{ROF} {RON}{PUR} {ROF}{ACL} {RON}{PUR}[2 ESP]{ROF}{
ACL} {BLU} {RON}[2 ESP]{PUR} {ROF}{ACL} {BLU} {RON}[2
 ESP]{PUR}[4 ESP]";                                   -089-
90 PRINT"{ROF}{ACL}[4 ESP]";                          -065-
91 PRINT"{PUR}[5 ESP]{RON} {BLU} {ROF}<ROF>{ACL}[2 ES
P]{BLU} {ACL} {RON}{PUR} {BLU} {ROF}<ROF>{RON}{PUR} {
BLU} {ROF}<ROF> {RON}{PUR} {ROF}{ACL}[2 ESP]{RON}{PUR
} {ROF}{ACL} {BLU} !BLU!{RON} {PUR} {ROF}{ACL} {BLU}
!BLU!{RON} {PUR} {ROF}{BLU}!BLU!";                    -073-
92 PRINT"{RON} {PUR} {ROF}{ACL}[4 ESP]";              -049-
93 PRINT"{ACL} {PUR} {ACL}[3 ESP]{RON}{PUR} {ROF}{BLU
}<BLU>{ACL}[2 ESP]{BLU} {ACL}[2 ESP]{RON}{PUR} {ROF}{
BLU}<BLU>{ACL} {RON}{PUR} {ROF}{BLU}<BLU>{ACL} {BLU}
{RON}{PUR} {ROF}{ACL}[2 ESP]{RON}{PUR} {ROF}{ACL} {BL
U} {ACL} {BLU}!BLU!{RON}{PUR} {ROF}{ACL} ";           -139-
94 PRINT"{BLU} {ACL} {BLU}!BLU!{RON}{PUR} {ROF}{ACL}
{BLU}!BLU!{RON}{PUR} {ROF}{ACL}[4 ESP]";              -034-
95 PRINT"{ACL}[40 ESP]";                              -052-
96 PRINT"{ACL}[40 ESP]";                              -053-
97 PRINT"{ACL}[12 ESP]{RON}{BLU}<BLU>!N!{ROF}{ACL}[2
ESP]{RON}{BLU}<BLU>!N!{ROF}{ACL}[2 ESP]{RON}{BLU}[2 E
SP]{ROF}{ACL}[2 ESP]{RON}{BLU}!M!!M!{ROF}{ACL} {RON}{
BLU}!M!!M!{ROF}{ACL}[11 ESP]";                        -077-
98 PRINT"{ACL}[11 ESP]{RON}{BLU}<BLU>!3 N!<N>!N! {ROF
}{ACL} {RON}{BLU}<M>[2 ESP]<G>{ROF}{ACL} {RON}{BLU} !
M!!M! !M!!M!{ROF}{ACL}[10 ESP]";                      -252-
99 PRINT"{ACL}[11 ESP]{RON}{PUR} {BLU}!3 N!{PUR} {BLU
}[2 ESP]{ROF} {RON}<M>{PUR}[2 ESP]{BLU} G {ROF}{ACL}
{RON}{BLU}[2 ESP]{PUR} {BLU}!M! {PUR} {ROF}{ACL}[10 E
SP]";                                                 -194-
100 PRINT"{ACL}[10 ESP]{PUR} {RON}[2 ESP]{BLU}!N!{PUR
}[2 ESP]{BLU}[2 ESP]{ROF}{ACL} {RON}{PUR} {ROF}{ACL}[
2 ESP]{RON}{PUR} {ROF}{ACL} {RON}{BLU}[2 ESP]{PUR}[2
ESP]{BLU}!M!{PUR} {ROF}{ACL}[10 ESP]";                -206-
101 PRINT"{ACL}[11 ESP]{RON}{PUR} {BLU} {PUR} {ROF}{A
CL} {RON}{PUR} {BLU}[2 ESP]{ROF}{ACL} {RON}{PUR} {BLU
}[2 ESP]{PUR} {ROF}{ACL} {RON}{BLU}[2 ESP]{PUR} {BLU}
 {PUR}[2 ESP]{ROF}{ACL}[10 ESP]";                    -047-
102 PRINT"";                                          -191-
103 PRINT"{ACL}[11 ESP]{RON}{PUR} {BLU} {ROF}<ROF>{AC
L} {RON}{PUR} {BLU} {ROF}<ROF> {RON}{PUR} {BLU}[2 ESP
]{PUR} {ROF}{ACL} {BLU}!BLU!{RON} {PUR} {ROF}{BLU}!BL
U!{RON} {PUR} {ROF}{ACL}[10 ESP]";                    -047-
104 PRINT"";                                          -193-
105 PRINT"{ACL}[11 ESP]{RON}{PUR} {ROF}{BLU}<BLU>{PUR
} {ACL} {RON}{PUR} {ROF}{BLU}<BLU>{ACL}[2 ESP]{BLU} {
RON}{PUR}[2 ESP]{ROF}{ACL}[2 ESP]{BLU} !BLU!{RON}{PUR
} {ROF}{ACL} {BLU}!BLU!{RON}{PUR} {ROF}{ACL}[8 ESP]";
                                                      -037-
106 PRINT"[2 ESP]";                                   -003-
107 PRINT"{ACL}[40 ESP]":                             -094-
108 PRINT"{ACL}[2 ESP]{WHT} {ACL}[2 ESP]{RON}{WHT}<B>
<2 I>{ROF}<F>{ACL} {WHT} <2 I><2 F><D>{ACL} {WHT}<D>
<O>{ACL} {WHT}<F>{ACL} {WHT}<I>{ACL} {WHT}<I><F>{ACL}
 {WHT}<D><F>{ACL}[10 ESP]";                           -120-
109 PRINT"{ACL}[5 ESP]{WHT}<K>{RON}<B>{ROF}<V><K>{ACL
} {WHT} {ACL} {WHT}<K>{ACL} {WHT}<K>{RON}<K>{ROF}{ACL
} {RON}{WHT}<K>{ROF}<B>{RON}<F>{ROF}{ACL} {WHT}<K>{RO
N}<K>{ROF}{ACL}[2 ESP]{RON}{WHT}<C><V>{ROF}{ACL} {WHT
}<K>{RON}<K>{ROF}{ACL}[4 ESP]";                       -207-
110 PRINT"[6 ESP]";                                   -126-
111 PRINT"{ACL}[2 ESP]{WHT} {ACL}[2 ESP]{WHT}<K><B><F
><K>{ACL} {WHT} {ACL} {WHT}<K>{ACL} {WHT}<B>{RON}<B>{
ROF}{ACL} {RON}{WHT}<K>{ROF}{ACL} {RON}{WHT}<K>{ROF}{
ACL} {WHT}<K><C><I>{ACL} {WHT}<K><B>{ACL} {WHT}<B {RO
N}<B>{ROF} COMMO";                                    -218-
112 PRINT"DORE";                                      -234-
113 PRINT"{ACL}[5 ESP]{WHT}<B><2 I><V>[30 ESP]{WHT} "
;                                                     -166-
114 PRINT"{ACL}[13 ESP]PULSA UNA TECLA'[10 ESP]{HOM}"
;                                                     -130-
115 POKE198,.:WAIT198,1:PRINT"{CLR}";:POKE198,.:GOTO8
                                                      -057-
116 PRINT"{CLR}";:FORI=PEEK(828)TOI+6:PRINTI:NEXT:PRI
NT"GOTO116"                                           -200-
117 POKE198,9:POKE631,19:FORI=632TO640:POKEI,13:NEXT
                                                      -194-
118 POKE828,PEEK(828)+6:IFPEEK(828)>115THENPRINT"{ARB
}[7 ESP]":POKE53248+21,.                              -174-
119 END                                               -146-
5000 PRINT"{CLR}{BLK}";:                              -078-
```





La revista semanal donde el usuario de COMMODORE podrá encontrar gran variedad de programas y noticias, sus periféricos y equipos.

SUSCRIPCION POR TELEFONO

Todos los días, excepto sábados y festivos, de **9 a 15,00 y de 17 a 19,00 horas** atenderemos sus consultas en el ☎ 2505820/2505579.
¡Suscríbase por teléfono! y en pocos días tendrá en su domicilio TU MICRO COMMODORE.

SUSCRIPCION POR CORREO

Deseo suscribirme a la revista TU MICRO COMMODORE al precio de 7.400 ptas. **(incluido IVA),** por el período de un año a partir del N.º....... inclusive.

El importe **—que abonaré en su totalidad con el primer envío—** lo haré efectivo de la siguiente forma:

☐ Talón bancario adjunto a nombre de INGELEK, S. A. ☐ Giro Postal n.º ....
☐ Tarjeta VISA n.º ______ ______ ______ ______ FIRMA
☐ Tarjeta MASTER CARD n.º ______ ______ ______ ______
Fecha de caducidad de la tarjeta .... Nombre titular ....

APELLIDOS ______ NOMBRE ______
DOMICILIO ______ EDAD ______
CIUDAD ______ PROVINCIA ______
C POSTAL ______ TELEFONO ______ PROFESION ______

Recorte o copie esta tarjeta y envíela dentro del sobre a: Ediciones INGELEK Apartado de Correos 61.294. 28080 MADRID

DISK NAME

Todavía no sabemos si los discos tienen o no tienen corazoncito, pero lo que si sabemos es que no son entes indefinidos, ya que cada uno, tiene su propio nombre.

En efecto, el proceso normal para «bautizar» a un disco, es a través del comando **NEW (OPEN15, :,15,"N: (OFF) ... etc),** que como todos sabemos, resulta ser un método totalmente destructivo de la información contenida en el disco hasta ese momento. Por medio de **NEW,** podemos, además, definir un código de dos caracteres, que se denomina identificación del disco; podemos decir que esto constituye el apellido del mismo.

Pero, si ya tenemos grabada información en el disco, y resulta que no la queremos perder, y además, deseamos cambiar el nombre al disco; ¿qué podemos hacer? Una solución sería formatear un nuevo disco y hacer un transvase de información de un disco a otro, pero lo mejor que puedes hacer es usar la rutina que presentamos. Con ella, se programa la unidad 1541, y aunque está desarrollada sobre un C64, no deberás tener problemas al usarla con cualquier ordenador Commodore (V20, C16, C128).

Como la rutina utiliza posibilidades poco conocidas de la unidad de disco 1541, vamos a explicarla línea por línea:

10: Se abren los ficheros números 1 y 2. El primero (1) se empleará para comunicar con el Sistema a través del canal de comandos (15), el segundo (2) abre un buffer en la unidad (el primero disponible).

20: A través del canal 15 (fichero 1), se comunica al Sistema que lea un bloque (u1), por el canal 2, del drive 0, y en concreto la pista 18, sector 0.

30: Una vez leído el bloque se posiciona el puntero del buffer (b-p) en el byte 144 del fichero 2. Aquí se encuentra la denominación del disco.

40: Se introduce el nombre del disco (16 caracteres o menos).

50: Se comprueba la longitud del nombre, si es superior a 16, entonces vuelve a preguntarlo.

60: Si el nombre no llega a 16 caracteres, se «rellena» con espacios con shift (160).

70: Se escribe en el fichero 2 (buffer), el nombre ya modificado.

80: Se efectúa una escritura del bloque situado en el buffer lógico 2 del drive 0, en la pista 18 sector 0 (es decir, se escribe el nuevo bloque con el nuevo nombre), y se cierra el 2.

90: Se inicializa la unidad para que se reconozca el nuevo nombre, y por último, se cierra el 1.

La unidad de disco 1541, tiene



Alterador del nombre del disco.

```
10 OPEN1,8,15:OPEN2,8,2,"#"                                -205-
20 PRINT#1,"U1 2 0 18 0"                                   -212-
30 PRINT#1,"B-P 2 144"                                     -158-
40 PRINT"NOMBRE DEL DISCO (MAX. 16 CARACTERES)":INPUT
A$                                                         -215-
50 IFLEN(A$)>16THEN40                                      -225-
60 IFLEN(A$)<16THENA$=A$+CHR$(160):GOTO60                  -112-
70 PRINT#2,A$;                                             -053-
80 PRINT#1,"U2 2 0 18 0":CLOSE2                            -189-
90 PRINT#1,"I0":CLOSE1                                     -052-
```

una cantidad de posibilidades de programación insospechadas. Desgraciadamente, sólo se vienen usando una mínima fracción de todos los comandos disponibles.

En la rutina comentada, se aprovechan algunas de estas cualidades más «tabú» como son:

Block Read (b-r) o (u1) o (ua).
Block Write (b-w) o (u2) o (ub).
Apertura de un buffer (256 bytes) de trabajo (#).
Buffer Pointer (b-p).

Una rutina parecida se puede emplear para leer solamente el nombre del disco. Esto se puede aprovechar en algún programa que necesite usar el disco, ya sea leer o escribir ficheros, por ejemplo. Para ello, nada mejor que la rutina de lectura del nombre, y así controlar que el disco que se ha introducido en la unidad es el adecuado.

Por ello, seamos imaginativos y demos a cada disco un nombre diferente. Ganaremos en claridad.



Visualizador del nombre del disco.

```
10 OPEN1,8,15:OPEN2,8,2,"#"                    -205-
20 PRINT#1,"U1 2 0 18 0"                       -212-
30 PRINT#1,"B-P 2 144"                         -158-
40 PRINT"NOMBRE DEL DISCO: ";                  -052-
50 GET#2,A$:IFASC(A$)=160THEN70                -011-
60 N$=N$+A$:GOTO50                             -015-
70 PRINTN$                                     -134-
80 CLOSE2                                      -048-
90 CLOSE1                                      -048-
```

C O N C U R S O

El travieso C-Byte tiene el honor de invitaros a la participación en nuestro tercer concurso de programación. Los requisitos necesarios son bien pocos:

- Saber programar un ordenador COMMODORE.
- Ser español o extranjero y
- Tener una edad comprendida entre 5 y 105 años.

Fácil, ¿verdad?
En cuanto a los premios, la mar de atractivos:

- **1.er premio.–60.000 pesetas en material informático a escoger por el premiado, más un póster de C-Byte.**
- **2.° premio.–30.000 pesetas en material informático a escoger por el premiado, más un póster de C-Byte.**
- **3.er premio.–15.000 pesetas en material informático a escoger por el premiado, más un poster de C-Byte.**

Y en fin, si alguno de los programas destaca por su originalidad, estética o comicidad, no sería de extrañar que le cayera alguna cosilla más...

Las bases del concurso son bien sencillas:

❶ Los programas remitidos al concurso deberán ser creación original del autor o autores, y completamente inéditos, pudiendo remitir tantos programas como se desee

❷ Los programas deberán ser enviados en casete o disco flexible a TU MICRO COMMODORE (Concurso de programación) Apartado de correos 61 294 28080 MADRID

❸ Los programas podrán ser de cualquier tipo (juegos, utilidades, gestión, educativos) y habrán de estar escritos en lenguaje BASIC o código máquina

❹ Los programas deberán ser remitidos desprovistos de cualquier tipo de protección, que impida o dificulte el análisis del mismo, así como reproducción en estas páginas y su introducción como listado siguiendo el sistema FUERA ERRORES.

❺ Cuando la ejecución del programa precise de la concurrencia de determinado periférico o aditamento (joysticks, tabletas gráficas, programas comerciales de ayuda), se valorará la indicación de las modificaciones pertinentes, para que el programa pueda ser disfrutado por cualquier usuario en la configuración básica

❻ Todo programa presentado al concurso deberá acompañarse de los siguientes datos

- Datos personales del concursante.
- Nombre del programa
- Modelo para el que está destinado
- Breve descripción del programa detallando las indicaciones necesarias para su ejecución

❼ Los programas premiados pasarán a ser propiedad de la revista TU MICRO COMMODORE, pudiendo hacer ésta libre uso de ellos, y renunciando sus autores a cualquier otra compensación distinta al premio

❽ Los programas no premiados que por su calidad se hagan merecedores de su publicación, serán adquiridos por la editorial, aplicando la tarifa vigente.

❾ Los programas recibidos con posterioridad a la fecha tope de la presente edición, serán automáticamente incluidos en los destinados a la siguiente

❿ El jurado decidirá sobre todos los aspectos no contemplados en estas bases y su decisión será inapelable

Y ahora a darse mucha prisa, el plazo para la recepción de programas termina el próximo día 15 de mayo (mes de las flores).

¡¡SUERTE!!





COMMODORE, COMPAÑIA AMERICANA, LIDER MUNDIAL EN NUMERO DE ORDENADORES INSTALADOS.



Commodore 128

Más 128 que nadie.

Commodore presenta el 128 más completo del mercado: El Commodore 128.

Un ordenador nacido para convertirse en mito.

Más prestaciones que nadie Para ser más que nadie hay que demostrar la capacidad de actuación. Para el C-128 estos son sus poderes:
• 122.365 Bytes libres en modo Basic • Biblioteca de programas más extensa del mercado (pues dispone de todos los programas del C-64, del C-128 y de CP/M® 3.0.). • Teclado numérico independiente • Alta resolución • 80 columnas en pantalla • Compatible con la periferia del C-64.

En una palabra, el ordenador más completo de la gama 128.

Más ordenador que nadie Además y por si fuera poco, el C-128 es el único ordenador capaz de actuar como tres.

Primero como un C-64, con cuyos programas y periféricos es compatible; segundo como un 128 en toda la extensión de la palabra; y tercero, como un ordenador con sistema operativo CP/M®.

Y todo, con sólo pulsar una tecla.

Ha comenzado la era de los 128, conózcalos y sepa que uno ya es más 128 que nadie, el C-128.

PRINCIPALES CARACTERISTICAS
Microprocesadores: 8502 (1 ó 2 MH_z); Z80A (4 MH_z); MMU para gestión de memoria. – RAM total de 128 Kb. – 122 Kb de RAM libres en modo BASIC. – ROM 48 Kb + 20 Kb. – Pantalla texto de 80 × 25 y 40 × 25. – Máxima resolución 640 × 200. – 16 colores y los 16 a la vez en pantalla. – 3 voces con control de envolvente y 8 octavas. – Teclado de 92 teclas con módulo numérico independiente.





Microelectrónica y Control *c/ Valencia, 49-53 08015 Barcelona - c/ Ardemans, 8 28028 Madrid*
Unico representante de Commodore en España.